A COMBINED WAVELET AND AUTOREGRESSIVE BASED STATISTICAL INTRUSION DETECTION SYSTEM
‘THE WAVELET-AR IDS”

by

Umut Güven

B.S., Electrical-Electronics Engineering, İstanbul University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2007

To my family
You are my hero
Thank you for all!

ACKNOWLEDGEMENTS

Firstly, I would like to thank to my thesis supervisor Prof. Dr. Emin Anarım for his kind interest, technical advices and his patience. I'm indebted to him for his tolerance and guidance.

Additionally, I would like to thank to Assist. Prof. Dr. Frédéric Kerem Harmancı for his kind interest, technical advices and his patience.

I would like to thank to my father Nebil Güven and my mother Müberra Güven for their support during my whole life.

I would like to thank to my friends Elif Tatlıdede, Mine Tatlıdede, and Suat Eyi for their help and support during my thesis study.

ABSTRACT

A COMBINED WAVELET AND AUTOREGRESSIVE BASED STATISTICAL INTRUSION DETECTION SYSTEM 'THE WAVELET-AR IDS"

Networks are complex interacting systems and are comprised of several individual entities such as routers and switches. Network performance information is not directly available, and the information obtained must be synthesized to obtain an understanding of the ensemble behavior. Threat from un-authorized users and remote attackers is increasing rapidly. There is a need for robust and reliable Intrusion Detection Systems. Common criterions for reliable IDS are low false positive rate and false negative rate, and high true positive rate and true negative rate. If IDS satisfies these criterions then it can be used to provide network security.

In this thesis, a new IDS scheme is proposed. Wavelet-AR IDS is designed to satisfy the criterions above. In the design phase the objective was to reduce to Autoregressive based IDS's false positive rate. The other objective was to design a new A operator matrix in order to increase the detection rate of Intrusion Detection System. The innovation in this thesis is to combine Wavelet and Autoregressive models in order to design a robust and reliable Intrusion Detection System. It is shown that Wavelet-AR IDS has acceptable false alarm rate and false negative rate, and Wavelet-AR IDS has high true positive rate and true negative rate. Consequently, we can say that Wavelet-AR IDS is a good Statistical Intrusion Detection System.

ÖZET

BİRLEŞTİRİLMİŞ DALGACIK VE ÖZBAĞLANIM TEMELLİ İSTATİSTİKSEL SALDIRI TESPİT SİSTEMİ "DALGACIK-ÖZBAĞLANIM SALDIRI TESPİT SİSTEMİ"

Bilgisayar ağları artan karmaşıklaşan yapılardır ve birçok bireysel ekipmandan oluşur, bunlar yönlendiriciler ve ağ anahtarlarıdır. Bilgisayar ağlarının verimlilik bilgisi doğrudan elde edilebilen bir veri değildir, elde edilen bilgiler bilgisayar ağlarının davranışlarını anlamak için analiz edilmelidir. Yetkisiz kullanıcıların ve uzak ağdaki saldırganların oluşturduğu tehditler giderek artmaktadır. Güvenilir, kararlı ve güçlü Saldırı Tespit Sistemlerine ihtiyaç vardır. Saldırı Tespit Sitemler için bazı genel ölçütler vardır. Bunlar, düşük yanlış-pozitif oranı ve yanlış-negatif oranı ile yüksek doğru-pozitif oranı ve doğru-negatif oranıdır. Eğer ki bir Saldırı Tespit Sitemi bu ölçütleri sağlarsa o zaman ağ güvenliği için kullanılabilir.

Bu tezde yeni bir Saldırı Tespit Sistemi tasarımı önerilmiştir. Dalgacık-Özbağlanım Saldırı Tespit Sistemi (STS) yukarıda belirtilen ölçütleri sağlayacak şekilde tasarlanmıştır. Tasarım evresinde, amaç Özbağlanım temelli STS'nin yanlış-pozitif oranını düşürmekti. Diğer bir amaç ise saldırı tespit oranını arttırmak için yeni bir A operatör matrisi tasarlamaktı. Bu tezdeki asıl yenilik ise Dalgacık ve Özbağlanım temelli modelleri güvenilir ve kararlı bir Saldırı Tespit Sistemi elde etmek için birleştirmekti. Gösterilmiştir ki Dalgacık-Özbağlanım STS kabul edilebilir yanlış-alarm oranı ve yanlış-negatif oranına sahiptir. Ve de Dalgacık-Özbağlanım STS yüksek doğru-pozitif oranına ve doğru-negatif oranına sahiptir. Sonuç olarak, Dalgacık-Özbağlanımlı STS iyi bir İstatistiksel Saldırı Tespit Sistemidir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

A	Operator matrix
A_{ip}	IP variables operator matrix
$A(\lambda,t)$	Average value of the signal $x(t)$
a	Vector of AR coefficients
B	Base rate
C	Covariance matrix
C_{ID}	Intrusion detection capability
$[c_{ij}]$	Covariance matrix
$c_{j,k}$	Approximated coefficients
d	Size of unitary matrix in SVD
$d_{j,k}$	Detailed coefficients
$D(\lambda,t)$	Difference between average values
ε_i	Residual error
e	Size of unitary matrix in SVD
$e_j(t)$	Residual of DWT
$E\{x\}$	Expected value of x
$f(\varepsilon_1,.....\varepsilon_i)$	Joint probability density function
g_n	High pass filter in DWT
h	AR decision threshold
h_n	Low pass filter in DWT
H_0	Hypothesis implying a no-change
H_1	Hypothesis implying a change
I	Identity matrix
ℓ	Likelihood ratio
L	Likelihood function
$L^2(R)$	Weighted spaces
μ	Mean of the segment

n	Abnormality vector length
N_R	Learning window length
N_S	Test window length
N'_S	$N_S - p$, test window length minus AR order
m	MA model order
M	Constant, determine the amount of overdetermination of ARMA
p	AR order
$p(\varepsilon_{N_R} / \alpha_p)$	Joint likelihood of the residual time series
PD_{avg}	Average of percentage deviation
PD_x	Percentage deviation of elements
$P(A)$	Probability of alarm
$P(I)$	Probability of intrusion
$R(t)$	Learning window
$r_i(t)$	Elements of learning window
$\tilde{r}_i(t)$	Estimation of $r_i(t)$
$r(i)$	Elements of covariance matrix of Yule Walker
R_n	Covariance matrix of Yule Walker
R_φ	Covariance matrix in SVD
$S(t)$	Test window
S	d-by-e matrix in SVD
t	Time
th	Threshold of decision
U	d-by-d unitary matrix
V	e-by-e unitary matrix
$\underline{v}$	Decorrelated abnormality vector
V_m	Subspaces
v	Elements of V
Y	Time series vector
W	MxM positive definite weighting matrix
W_m	Orthogonal complement of spaces

$W(\lambda,t)$	Wavelet transform, *t*: time, λ: scale
Σ	*d*-by-*e* with nonnegative numbers
x	General variable
$x(t)$	Real valued function
X	Input vector of "corrcoef" matlab function
2^j	Scale number of wavelet transform
α	AR parameters or AR coefficients
α_R	AR coefficients learning window segment
α_S	AR coefficients test window segment
σ^2	Variance of segment
σ_R^2	Variance of learning window segment
σ_S^2	Variance of test window segment
σ_p^2	Variance of pooled window segment
η	Likelihood ratio
η_L	Log likelihood ratio
η_{IR}	Likelihood ratio of IPinreceives
η_{IDe}	Likelihood ratio of IPindelivers
$\eta_{OR.}$	Likelihood ratio of IPoutrequest
λ	Scale of wavelet transform
λ_d	Decision Variable of AR model
θ	Vector of AR coefficients in Yule-Walker
$\hat{\gamma}_k$	Variance estimate of the Modified Yule-Walker method
$\psi(t)$	Mother wavelet function or basis function
$\varphi(t)$	Abnormality vector
$\phi(x)$	Scaling filter

AC	Autocorrelation
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ASN	Abstract Syntax Notation
CD	Compact Disc
CPM	Change-Point Monitoring
CPU	Central Processing Unit
CUSUM	Cumulative Sum
CWT	Continuous Wavelet Transform
DARPA	Defense Advanced Research Projects Agency
DFT	Discrete Fourier Transform
DNS	Domain Name Service
DDoS	Distributed Denial of Service
DoS	Denial of Service
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
GLR	Generalized Likelihood Ratio
HIDS	Host based Intrusion Detection System
IDS	Intrusion Detection System
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IRC	Internet Relay Chat
IT	Information Technology
KB	Kilobyte
KDD	Knowledge-Discovery in Databases
LLR	Log likelihood Ratio
LS	Least Squares
MA	Moving Average
MFLOP	Millions of Floating Operations per Second

MIB	Management Information Base
MODWT	Maximal Overlap Discrete Wavelet Transform
NIDS	Network based Intrusion Detection System
NPV	Negative Predictive Value
NT	New Technology
OD	Origin Destination
OS	Operating System
PAYL	Payload Based Anomaly Detector
PC	Personal Computer
PCA	Principal Components Analysis
PPV	Positive Predictive Value or Bayesian Detection Rate
SNMP	Simple Network Management Protocol
SVD	Singular Value Decomposition
SYN	Synchronize Packet
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol

1. INTRODUCTION

Networks are complex interacting systems and are comprised of several individual entities such as routers and switches. The behavior of the individual entities contributes to the ensemble behavior of the network. The evolving nature of Internet Protocol (IP) networks makes it difficult to fully understand the dynamics of the system [1].

1.1. Network Security

To obtain a basic understanding of the performance and behavior of these complex networks, huge amounts of information need to be collected and processed. Often, network performance information is not directly available, and the information obtained must be synthesized to obtain an understanding of the ensemble behavior [1]. Using some basic knowledge of the network layout as well as the traffic characteristics at the individual nodes, it is possible to detect network anomalies and performance bottlenecks. The detection of these events can then be used to trigger alarms to the network management system [1].

Network-based data pertains to the functioning of the network devices themselves and includes information gathered from the router's and switches physical interfaces as well as from the router's forwarding engine. Traffic counts obtained from both types of data can be used to generate a time series to which statistical signal processing techniques can be applied [1].

However, in some cases, only descriptive information such as the number of open TCP (Transmission Control Protocol) connections, source-destination address pairs, and port numbers are available. In such situations, conventional approaches of rule-based methods would be more useful. The existing network management schemes use thresholds to generate alarms. These thresholds are set based on the expertise of a human network manager. Such systems cannot reliably detect impending network problems [2].

Several network management software packages are commercially available; however these at best can only detect severe failures or performance issues such as a broken link or a loss of link capacity. These methods do not capture changes in network traffic that are indicative of many common network problems including file server crashes. Rule based methods have also been developed to detect certain subsets of faults. Unfortunately these methods require a data base of fault scenarios and rules for detection which often rely heavily on the expertise of the network manager. The rules thus developed are too specific to characterize all network fault scenarios that evolve with time. Thus most schemes based on Artificial Intelligence suffer from being dependent on prior knowledge about the fault conditions on the network and the rules developed do not adapt well to a changing network environment [3].

The information obtained at the router and switches is the aggregate of the information from all the subnets. The router, which is primarily a network layer device, processes the IP layer information which is a multiplexing of traffic from all of the interfaces. This distributed scheme allows for problem isolation to a specific sub-network. The Intelligent Agent is a processing algorithm much like a software entity that has as its inputs the MIB (Management Information Base) variables that are specific to the router and its output provides a parameter that is a predictive indicator of network health [4].

1.1.1. What is a Intrusion Detection System?

Intrusion detection is the process of monitoring networks for unauthorized access, activity or file modification. Intrusion Detection System can also be used to monitor network traffic, thereby detecting if a system target by an attack [33]. In other words, an Intrusion Detection System is a defense system, which detects hostile activities or exploit in a network. The key is then to detect and if possible prevent activities that may compromise system security or hacking attempt in progress. One key feature of intrusion detection system is their ability to provide a view of unusual activity and issue alerts notifying administrator [33].

As far as monitoring approach and securing data is concerned, there are two basic types of IDS: Host based IDS and Network based IDS. In short, Host based Intrusion

Detection Systems (HIDS) are placed on host machines and examine the activities and access to key serves upon which a HIDS has been placed. Network based Intrusion Detection Systems (NIDS) examines the packet passing through the network. There are several techniques to detect intrusions. According the detection approaches, existing intrusion detection schemes can be further divided in two classes; Anomaly Detection and Misuse Detection. Misuse detection system systems try to match computer activity to stored signatures of known exploits or attacks while anomaly detection system first learns the normal system behavior and looks for deviation from that. We are mainly focus on Network based IDS and a different way of Anomaly Detection method, because we do not need to run the method firstly on normal network environment.

1.2. Attack Definitions and Types

It's impossible to explain all types of network attacks exist in today's computer network technology. In this part the types of attacks that can be detected SNMP (Simple Network Management Protocol) data are mentioned.

Generally, attacks can be categorized in two areas as passive and active attacks. Passive attacks are aimed at gaining access to penetrate the system without compromising IT (Information Technology) resources. Active attacks result in an unauthorized state change of IT resources. In terms of the relation intruder-victim, attacks are categorized as, internal and external attacks. Internal attacks come from own enterprise's employees or their business partners or customers. External attacks come from outside of the enterprise network, frequently via the Internet. Attacks are also identified by the source category, namely those performed from internal systems (local network), the Internet or from remote dial-in sources [29].

The attack types that can be detectable (sometimes hardly detectable) by IDS systems can be listed as follows [29].

- Those related to unauthorized access to the resources can be listed as follows.
- Password cracking and access violation.

- Trojan horses.
- Interceptions; most frequently associated with TCP/IP stealing and interceptions that often employ additional mechanisms to compromise operation of attacked systems (for example by flooding); man in the middle attacks).
- Spoofing (deliberately misleading by impersonating or masquerading the host identity by placing forged data in the cache of the named server i.e. DNS (Domain Name Service) spoofing).
- Scanning ports and services, including ICMP (Internet Control Message Protocol) scanning (Ping), UDP, TCP Stealth Scanning TCP that takes advantage of a partial TCP connection establishment protocol.).
- Remote OS (Operating System) Fingerprinting, for example by testing typical responses on specific packets, addresses of open ports, standard application responses (banner checks), IP stack parameters etc.,
- Network packet listening (a passive attack that is difficult to detect but sometimes possible),
- Stealing information, for example disclosure of proprietary information,
- Authority abuse; a kind of internal attack, for example, suspicious access of authorized users having odd attributes (at unexpected times, coming from unexpected addresses).
- Unauthorized network connections,
- Usage of IT resources for private purposes, for example to access pornography sites,
- Taking advantage of system weaknesses to gain access to resources or privileges,
- Unauthorized alteration of resources (after gaining unauthorized access):
- Falsification of identity, for example to get system administrator rights,
- Information altering and deletion,
- Unauthorized transmission and creation of data (sets), for example arranging a database of stolen credit card numbers on a government computer (e.g. the spectacular theft of several thousand numbers of credit cards in 1999),

- Unauthorized configuration changes to systems and network services (servers).

From the above list, only the attack types that causes an abnormal change in the traffic variables (such as bandwidth, link utilization, number of incoming, outgoing or erroneous packets) can be detected with SNMP data. Because of this reason denial of service (DoS) attacks can be detected with SNMP data.

1.2.1. Types of Denial of Service Attacks

Denial of service attacks are generally based on ICMP (Internet Control Message Protocol) Floods, Smurf Attacks (which are also ICMP floods, but uses the broadcast address), UDP (User Datagram Protocol) Flood, TCP (Transmission Control Protocol) Flood, TCP SYN Flood, Spoofing (by falsifying the IP address and attacking), Ping of Death (pretty much outdated now), Application Attack (attacking a vulnerability in an application), Teardrop (IP fragmentation, again pretty much outdated now), Fragile Attack (which is similar in nature to a Smurf Attack, except it uses UDP as opposed to TCP) [30]. DoS attacks can be classified into two main categories:

1.2.1.1. Logic or software attacks: A small number of malformed packets are designed to exploit known software bugs on the target system. These attacks are relatively easy to counter either through the installation of software patches that eliminate the vulnerabilities or by adding specialized firewall rules to filter out malformed packets before they reach the target system.

These attacks do not affect traffic variables that can be obtained by SNMP protocol, so these attacks are not mentioned in this document. These attacks can be used detected signature based detection.

1.2.1.2. Flood attacks : A remote system is overwhelmed by a continuous flood of traffic designed to consume resources at the targeted server (CPU cycles and memory) and/or in the network (bandwidth and packet buffers). These attacks result in degraded service or a complete site shutdown.

TCP SYN Flood Attack; Taking advantage of the flaw of TCP three-way handshaking behavior, an attacker makes connection requests aimed at the victim server with packets with unreachable source addresses. The server is not able to complete the connection requests and, as a result, the victim wastes all of its network resources. A relatively small flood of bogus packets will tie up memory, CPU (Central Processing Unit), and applications, resulting in shutting down a server.

Smurf IP Attack; An attacker sends forged ICMP echo packets to broadcast addresses of vulnerable networks. All the systems on these networks reply to the victim with ICMP echo replies. This rapidly exhausts the bandwidth available to the target, effectively denying its services to legitimate users.

UDP Flood Attack; UDP is a connectionless protocol and it does not require any connection setup procedure to transfer data. A UDP Flood Attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no application that is waiting on the port, it will generate an ICMP packet of destination unreachable to the forged source address. If enough UDP packets are delivered to ports on victim, the system will go down.

ICMP Flood Attack; An ICMP attack can come in many forms. There are two basic kinds, Floods and Nukes. An ICMP flood is usually accomplished by broadcasting either a bunch of pings (Not IRC (Internet Relay Chat) pings, ICMP pings. Similar purpose, but handled differently) or UDP packets (which are used in software like PointCast). The idea is, to send so much data to your system, that it slows you down so much that you're disconnected from IRC due to a ping timeout. Nukes exploit bugs in certain Operating

systems, Like Windows 95, and Windows NT (New Technology). The idea is to send a packet of information that the OS can't handle. Usually, they cause your system to lock up.

Denial of Service (DoS) Type Attacks:

- Flooding – compromising a system by sending huge amounts of useless information to lock out legitimate traffic and deny services:
- Ping flood (Smurf) – a large number of ICMP packets sent to a broadcast address,
- Send mail flood - flooding with hundreds of thousands of messages in a short period of time; also POP and SMTP relaying,
- SYN flood – initiating huge amounts of TCP requests and not completing handshakes as required by the protocol,
- Distributed Denial of Service (DDoS); coming from a multiple source,
- Compromising the systems by taking advantage of their vulnerabilities:
- Buffer Overflow, for example Ping of Death — sending a very large ICMP (exceeding 64 KB),
- Remote System Shutdown,
- Web Application attacks; attacks that take advantage of application bugs may cause the same problems as described above

1.3. SNMP Protocol Overview

This section contains an overview of SNMP (Simple Network Management Protocol). SNMP is a communication protocol that has gained widespread acceptance since 1993 as a method of managing TCP/IP networks. SNMP was developed by the IETF (Internet Engineering Task Force), and is applicable to any TCP/IP network, as well as other types of networks. The protocol has been in existence for some time, and has been written about extensively [31].

SNMP defines a client/server relationship. An SNMP agent is software that resides on a network node and is responsible for communicating with managers regarding that node. The node is represented as a managed object having various fields or variables that are defined in the appropriate MIB. The MIB (Management Information Base) is a method of describing managed objects by specifying the names, types, and order of the fields (or variables) that make up the object. The MIB can either be a standard one or can be what is known as an enterprise MIB.

The client (network manager) makes virtual connections to a server (SNMP agent), which executes on a remote network device, and serves information to the manager regarding the device's status. The database, controlled by the SNMP agent, is referred to as the SNMP Management Information Base (MIB), and is a standard set of statistical and control values. SNMP additionally allows the extension of these standard values with values specific to a particular agent through the use of private MIBs.

Directives, issued by the network manager client to an SNMP agent, consist of the identifiers of SNMP variables (referred to as MIB object identifiers or MIB variables) along with instructions to either get the value for the identifier, or set the identifier to a new value.

Through the use of private MIB variables, SNMP agents can be tailored for a many specific devices, such as network bridges, gateways, and routers. The definitions of MIB variables supported by a particular agent are incorporated in descriptor files, written in Abstract Syntax Notation (ASN.1) format, made available to network management client programs so that they can become aware of MIB variables and their usage.

SNMP has several advantages. Its biggest strength is arguably its widespread popularity. SNMP agents are available for network devices ranging from computers, to bridges, to modems, to printers. The fact that SNMP exists with such support gives considerable credence to its reason for existence; SNMP has become interoperable.

Additionally, SNMP is a flexible and extensible management protocol. Because SNMP agents can be extended to cover device specific data, and because a clear

mechanism exists for upgrading network management client programs to interface with special agent capabilities (through the use of ASN.1 files), SNMP can take on numerous jobs specific to device classes such as printers, routers, and bridges, thereby providing a standard mechanism of network control and monitoring

1.3.1. Choice of Variable

To be compatible with the existing standards, the intelligent agent uses the standard MIB variables as its input parameters. These variables are supported by the current SNMP framework. By appropriately choosing the MIB variables that are representative of traffic flow at a node, the intelligent agent is capable of generalizing to heterogeneous nodes. Furthermore, MIB variables are supported by most network devices, thus making widespread application of the agent feasible [2].

The Management Information Base variables (MIB II), which are standardized for the Simple Network Management Protocol (SNMP) version (1), fall into different groups. The Internet Protocol (*ip*) group variables were determined sufficient to describe the functionality of the router and switches. The variables used in the intelligent agent represent cross sections of the traffic at different points in the *ip* layer. The variables *ipIR* (InReceives) represents the total number of datagram received from all interfaces of the router, *ipIDe* (InDelivers) represents the number of datagram correctly delivered to the higher layers, as this node was their final destination, and *ipOR* (OutRequests) represents the number of datagram passed on from the higher layers of the node to be forwarded by the *ip* layer. The MIB variables chosen, although non-redundant, are not strictly independent and the relationships between them have been incorporated at the combination stage described in [4].

As described in [32], *If_InOctets* and *If_OutOctets* variables can also be used for anomaly detection. And as we have shown in Performance Measurements part the interface variables contributes the decision phase in positive way. In some cases they have increased the detection ratio. *If_InOctets;* The traffic going into that interface from the router, *If_OutOctets*; The traffic out of that interface from the router.

1.4. Motivation

The goal of this work is to show the potential to apply signal processing techniques to the problem of network anomaly detection. Application of such techniques will provide better insight for improving existing detection tools as well as provide benchmarks to the detection schemes employed by these tools. Rigorous statistical data analysis makes it possible to quantify network behavior and, therefore, more accurately describe network anomalies [1]. The scope of this work is to describe the problem of IP network anomaly detection in a single administrative domain along with the types and sources of data available for analysis. Special emphasis is placed on motivating the need for signal processing techniques to study this problem. We present a technique based on Wavelet-AR change detection for addressing this challenge.

Furthermore, there is no single variable or metric that captures all aspects of normal network function. This presents the problem of synthesizing information from multiple metrics, each of which has widely differing statistical properties. To address this issue, we use an operator matrix to correlate information from individual metrics [1].

1.5. Thesis Outline

The rest of the thesis is organized in the following fashion. Section two makes introduction to Anomaly Detection Methods. In section three Autoregressive based model description is explained. In section four Wavelet-AR model has been proposed. In section five results of Autoregressive model has been showed. In section six results of Wavelet-AR model has been showed. In section seven Performance Measurements have been showed. Section eight concludes the thesis.

2. ANOMALY DETECTION METHODS

In this section, we review the most commonly used network anomaly detection methods. The methods described are rule-based approaches, pattern matching, Wavelet Models, and statistical analysis.

2.1. Rule Based Models

Early work in the area of fault or anomaly detection was based on expert systems. In expert systems, an exhaustive database containing the rules of behavior of the faulty system is used to determine if a fault occurred. Rule-based systems are too slow for real-time applications and are dependent on prior knowledge about the fault conditions on the network [1]. Furthermore, the identification of relevant criteria for the different faults will, in turn, require a set of rules to be developed. In addition, using any functional approximation scheme, such as back propagation, causes an increase in computation time and complexity. The number of functions to be learned also increases with the number of faults studied [1].

Alhamaty et al concentrate on finding a solution to the intrusion detection main attacks of fragmentation information packets. Main idea is to check TCP packet integrity so as not to restrict the check attack special signature. This work focuses on the packet that whether packet rightly is fragmented or not, until by change attack signature [18].

2.2. Pattern Matching Models

The efficiency of this pattern matching approach depends on the accuracy of the traffic profile generated. Given a new network, it may be necessary to spend a considerable amount of time building traffic profiles. In the face of evolving network topologies and traffic conditions, this method may not scale gracefully [1]. The methods of data analysis and pattern recognition presented are the basis of a technology study for an automatic intrusion detection system that detects the attack in the reconnaissance stage [22].

2.3. Multiscale and Multidimensional Analysis

These wavelet-based scaling analysis tools are incredibly useful for describing and detecting certain kinds of properties of one-dimensional functions, measures, or random processes. They can compute summary statistics about scale-dependent properties, local scaling behavior, and even extremely localized information about the local regularity of network traffic. Because these methods can be implemented in an on-line fashion, thet use them to monitor network links, either at one main link or at many access points. However, these tools do have some serious drawbacks when it comes to the next step in network measurements. A. C. Gilbert works only with information local to a network. Gilbert cannot address distributed network measurements at all [17].

An approach for real-time network monitoring in terms of numerical time-dependant functions of protocol parameters was suggested in [22]. *Gudkov et al* have applied complex systems theory for information flow analysis of networks, the information traffic is described as a trajectory in multi-dimensional parameter-time space with about 10-12 dimensions. The network traffic description is synthesized by applying methods of theoretical physics and complex systems theory, to provide a robust approach for network monitoring that detects known intrusions, and supports developing real systems for detection of unknown intrusions [22].

2.4. Wavelet Models

The link loads and traffic matrices are simply related by a linear equation $b = Ax$ The vector b contains the link measurements, and A is the routing matrix. They wish to infer x, which contains the unknown traffic matrix elements written as a vector. Tomographic inference techniques seek to invert this relationship to find x. Two basic solution strategies to network anomography: (i) *early inverse*, and (ii) *late inverse*. Early inverse approaches may appear more intuitive. The early inverse approach tackles the problem in two steps. The first is the *network tomography* step, where OD (Origin Destination) flow data at each interval j are inferred from the link load measurements by solving the ill-posed linear inverse problem. Given the estimated OD flow data x_j at different time points j, in the second step, *anmaly detection* can then be applied to the x_j [19].

In [19] *Zhang et al* have showed that the ARIMA (Autoregressive Integrated Moving Average) methods, FFT (Fast Fourier Transform) and Wavelet anomography approaches have superb performance the number of false negatives is very low. This indicates that very few important traffic anomalies can pass undetected by these approaches. The PCA based approaches, however, identify about half of the anomalies [19].

In [20], *Huang et al* apply signal processing techniques in intrusion detection systems, and develop and implement a framework, called Waveman, for real time wavelet-based analysis of network traffic anomalies. Then, they use two metrics, namely percentage deviation and entropy, to evaluate the performance of various wavelet functions on detecting different types of anomalies like Denial of Service (DoS) attacks and portscans. Results show that Coiflet and Paul wavelets perform better than other wavelets in detecting most anomalies considered in this work [20].

Inspired by the methods that use the selfsimilarity property of data network traffic as normal behavior and any deviation from it as the anomalous behavior, In [21], *Rawat et al* have proposed a method for anomaly based network intrusion detection. Making use of the relations present among the wavelet coefficients of a self-similar function in a different way, method determines the possible presence of not only an anomaly, but also its location in the data. They provide the empirical results on KDD (Knowledge-Discovery in Databases) data. Hurts parameter was used to perform anomaly detection [21].

In [23], *Kim et al* suggest a technique for traffic anomaly detection based on analyzing correlation of destination IP addresses in outgoing traffic at an egress router. This address correlation data are transformed through discrete wavelet transform for effective detection of anomalies through statistical analysis. Results from trace-driven evaluation suggest that proposed approach could provide an effective means of detecting anomalies close to the network [23]. Based on statistical bounds on normal traffic patterns of the correlation signal of destination addresses, sudden changes can be used to detect anomalies in traffic behavior. A correlation calculation is using a simple data structure. These correlation data are processed through coefficient selective discrete wavelet transform for effective and high-confidence detection [24].

2.5. Image Based Anomaly Detection

In [25] *NetViewer* was introduced a network measurement approach that can simultaneously detect, identify and visualize attacks and anomalous traffic in real-time by passively monitoring packet headers. *Kim et al* propose to represent samples of network packet header data as frames or images. With such a formulation, a series of samples can be seen as a sequence of frames or video, revealing certain kinds of attacks to the human eye. This enables techniques from image processing and video compression to be applied to the packet header data to reveal interesting properties of traffic. They show that "scene change analysis" can reveal sudden changes in traffic behavior or anomalies. They also show that "motion prediction" techniques can be employed to understand the patterns of some of the attacks. They show that it may be feasible to represent multiple pieces of data as different colors of an image enabling a uniform treatment of multidimensional packet header data [25].

Their approach passively monitors packet headers of network traffic at regular intervals and analyzes the aggregate data for anomaly detection. Their approach generates images of the packet header data for both visualization and for effective processing of the collected data. During network anomalies or attacks, the usage pattern of network may change and the peculiarities could become visible in the traffic images. When anomalies are detected, further analysis can characterize the anomalies by their nature into several categories and help in mitigating the attacks [25].

2.6. Change Point Detection

Wang et al present a simple mechanism, called Change-Point Monitoring (CPM), to detect denial of service (DoS) attacks. The core of CPM is based on the inherent network protocol behaviors and is an instance of the Sequential Change Point Detection. To make the detection mechanism insensitive to sites and traffic patterns, a nonparametric Cumulative Sum (CUSUM) method was applied, thus making the detection mechanism robust, more generally applicable, and its deployment much easier [26].

CPM compares the observed sequence with the profile that represents the user's normal behavior and detects any significant deviation from the normal behavior. The key difference of CPM from others is that CPM exploits the inherent network protocol behaviors, instead of traffic patterns, for detecting network anomalies. The objective of Change-Point Detection is to determine if the observed time series is statistically homogeneous and, if not, to find the point in time when the change happens [26].

2.7. PCA

Labib et al have proposed a multivariate statistical method called Principal Component Analysis is used to detect Denial-of-Service and Network Probe attacks using the 1998 DARPA (Defense Advanced Research Projects Agency) data set. Visualization of network activity and possible intrusions is achieved using Bi-plots, which are used as a graphical means for summarizing the statistics. The principal components are calculated for both attack and normal traffic, and the loading values of the various feature vector components are analyzed with respect to the principal components. The variance and standard deviation of the principal components are calculated and analyzed. A brief introduction to Principal Component Analysis and the merits of using it for detecting the selected intrusions are discussed [27].

2.8. PAYL Model

Bolzoni et al have proposed POSEIDON which is payload-based, and has a two-tier architecture: the first stage consists of a Self-Organizing Map, while the second one is a modified PAYL (Payload Based Anomaly Detector) system. Their architecture combines a SOM with a modified PAYL algorithm. POSEIDON, like most network intrusion detection systems, is packet-oriented. This architecture presents two main advantages: firstly, POSEIDON can identify and block an attack while it is taking place (intrusion prevention). Secondly, connection-based systems are computationally more expensive, in particular they require a huge amount of memory resources to keep all the segments to analyze. This makes connection-based system more suitable for off-line analysis [28].

Bolzoni et al have proposed APHRODITE which is an architecture designed to reduce false positives in network intrusion detection systems. APHRODITE works by detecting anomalies in the output traffic, and by correlating them with the alerts raised by the NIDS working on the input traffic. Benchmarks show a substantial reduction of false positives and that APHRODITE is effective also after a "quick setup", i.e. in the realistic case in which it has not been "trained" and set up optimally. APHRODITE works as follows: when the NIDS raises an alert, the correlation engine checks whether the communication that raised this alert also causes an anomaly in the output (detected by the OAD (Output Anomaly Detector)). If this is the case, the alert is considered a true positive and APHRODITE forwards it to the IT professionals, otherwise, it is discarded as a false positive [16]. They have tested APHRODITE together with both POSEIDON and Snort to on the traffic of weeks 4 and 5 of DARPA.

2.9. Statistical Analysis Models

Using online learning and statistical approaches, it is possible to continuously track the behavior of the network. Statistical analysis has been used to detect both anomalies corresponding to network failures, as well as network intrusions [1].

Qingtao et al has presented a method of detecting network anomalies by analyzing the abrupt change of time series data obtained from Management Information Base (MIB) variables. The method applies the Auto-Regressive (AR) process to model the abrupt change of time series data, and performs sequential hypothesis test to detect the anomalies [32].

2.10. Comments on Related Works

Signature based models and patter matching models can not detect newly designed attacks because of lack of information of new attacks types. They need continuous updates to catch the up to date attacks. Using case-based reasoning for describing fault scenarios also suffers from heavy dependence on past information. Furthermore, the identification of relevant criteria for the different faults will, in turn, require a set of rules to be developed. In addition, using any functional approximation scheme, such as back propagation, causes

an increase in computation time and complexity. The number of functions to be learned also increases with the number of faults studied.

We have performed a Wavelet based analysis on packet counter base information this work show us that only applying wavelet analysis and some threshold methods can not perform a sensitive analysis like AR based models. We have given some examples of payload based models they work offline and spend some much CPU power to reach results.

3. METHODOLOGY - AR

3.1. Least Squares AR(1) Estimate

In statistical analysis, a network anomaly is modeled as correlated abrupt changes in network data. An abrupt change is defined as any change in the parameters of a time series that occurs on the order of the sampling period of the measurement. Abrupt changes in time series data can be modeled using an autoregressive (AR) process [1]. *Totthan et al* have suggested a statistical analysis method to detect abrupt changes based on AR(1) LS.

Once the appropriate set of MIB variables were chosen, variable level alarms were obtained using a change detection algorithm [3]. It has been experimentally shown that changes in the statistics of traffic data can be used to detect faults [3]. The detection algorithm was implemented independently on each MIB variable.

The increments in the MIB counters were obtained every a second and the data thus generated constituted a time series. Note that the data exhibits a high degree of nonstationarity. Piecewise stationary Autoregressive models have been used to successfully describe such nonstationary stochastic time series signals [3].

Thus the MIB data were divided into 10 time lags piecewise stationary windows. Within a time window of size N ($N=10$), the MIB data was linearly modeled using a first-order AR process. Using these piecewise stationary windows. Piecewise stationary segments $R(t)$ and $S(t)$ shown in Figure 3.1. $R(t)$ is learning window and $S(t)$ is test window. Non overlapping windows were used in order to obtain less correlated residuals. $N_R=N_S=10$.

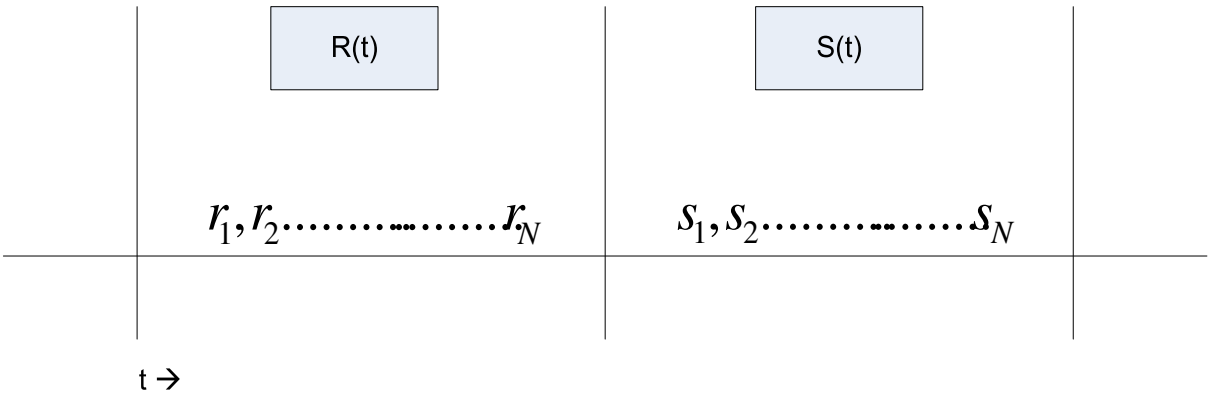

Figure 3.1. Piecewise stationary segments

Lets define *R(t)*;

$$R(t) = \{r_1(t), r_2(t), \ldots\ldots\ldots, r_{N_R}(t)\} \tag{3.1}$$

Here we can state any $r_i(t)$ as $\tilde{r}_i(t)$, where $\tilde{r}_i(t) = r_i(t) - \mu$ and μ is the mean of the segment *R(t)*. Now $\tilde{r}_i(t)$ can be estimated as an AR order *p* process (*p=1*) with a residual error ε_i;

$$\varepsilon_i(t) = \sum_{k=0}^{p} \alpha_k \tilde{r}(t-k) \tag{3.2}$$

Where $\alpha_R = \{\alpha_1, \alpha_2, \ldots\ldots, \alpha_p\}$ are the AR parameters, and $\varepsilon_i(t)$ is assumed to be white noise. The joint probability density function of $\varepsilon_1(t), \varepsilon_2(t), \ldots\ldots, \varepsilon_i(t)$ is given by [5],

$$f(\varepsilon_1, \ldots\ldots \varepsilon_i) = (2\pi\sigma^2)^{-N/2} \exp\{-\frac{1}{2\sigma^2} \sum_{t=1}^{N} \varepsilon_t^2\} \tag{3.3}$$

Substituting for $\varepsilon_i(t)$ from (3.3) and making use of the $x_{1-p}, x_{2-p}, \ldots\ldots x_0$ the likelihood function *L* of the { $\varepsilon_i(t)$ } is

$$L = (2\pi\sigma^2)^{-N/2} \exp\{-\frac{1}{2\sigma^2} \sum_{t=1}^{N} (\sum_{i=0}^{p} \alpha_i \chi_{t-i})^2\} \tag{3.4}$$

Which can rewritten as,

$$L = (2\pi\sigma^2)^{-N/2} \exp\{-\frac{1}{2\sigma^2} Na'Ca\} \tag{3.5}$$

Where a is the column vector given by, $a' = [1, \alpha_1, ..., \alpha_p]$ and C=[c_{ij}] is the $(p+1)x(p+1)$ matrix of covariance given by,

$$c_{ij} = \frac{1}{N}\sum_{t=1}^{N} \chi_{t-i}\chi_{t-J} \qquad i, j = 0,1,...., p \tag{3.6}$$

To obtain the maximum-likelihood estimates of σ^2 and α we must maximize L with respect to σ^2 and α. *Peter et al* show that this leads to the estimates σ^2 and α where $\tilde{\sigma}$ (**covariance estimate**)

$$\hat{\sigma}^2 = a'Ca \tag{3.7}$$

When we turn back to our notation, error is $N(0, \sigma_R^2)$ distribution. Joint likelihood of the residual time series was obtained as;

$$p(\varepsilon_{p+1}, ..., \varepsilon_{N_R} / \alpha_1, ..., \alpha_p) = \left(\frac{1}{\sqrt{2\pi\sigma_R^2}}\right)^{N'_R} \exp\left(\frac{-N'_R\hat{\sigma}_R^2}{2\sigma_R^2}\right) \tag{3.8}$$

Where σ_R^2 is the variance of residual in segment *R(t)* and $N'_R = N_R - p$ and $\hat{\sigma}_R$ is the covariance estimate of σ_R^2 [3]. Joint likelihood L of the residuals $R(t)$ and $S(t)$ is;

$$l = \left(\frac{1}{\sqrt{2\pi\sigma_R^2}}\right)^{N'_R} \left(\frac{1}{\sqrt{2\pi\sigma_S^2}}\right)^{N'_S} \exp\left(\frac{-N'_R\hat{\sigma}_R^2}{2\sigma_R^2}\right) \exp\left(\frac{-N'_S\hat{\sigma}_S^2}{2\sigma_S^2}\right) \tag{3.9}$$

And $N'_S = N_S - p$, σ_S^2 is variance of the residual in the segment *S(t)*. Two hypotheses are H_0 implying that no change, H_1 implying a change. Under the hypothesis H_0 we have; $\alpha_R = \alpha_S$ and $\sigma_R^2 = \sigma_S^2 = \sigma_p^2$ where σ_P^2 is the pooled variance.

$$l_p = \left(\frac{1}{\sqrt{2\pi\sigma_p^2}} \right)^{N'_R + N'_S} \exp\left(\frac{-(N'_R + N'_S)\hat{\sigma}_P^2}{2\sigma_P^2} \right) \tag{3.10}$$

Under the hypothesis H_1 we have; $\alpha_R \neq \alpha_S$ and $\sigma_R^2 \neq \sigma_S^2$ and under hypothesis H_0 $\hat{\sigma}_R^2 = \sigma_R^2$ and $\hat{\sigma}_S^2 = \sigma_S^2$ using conditions we obtained the likelihood ratio as;

$$\ell = \sigma_P^{-(N'_R + N'_S)} \sigma_R^{N'_R} \sigma_S^{N'_S} * \exp\left(\frac{-\hat{\sigma}_P^2 (N'_R + N'_S)}{2\sigma_P^2} + \frac{1}{2}\left[\frac{N'_R \hat{\sigma}_R^2}{\sigma_R^2} + \frac{N'_S \hat{\sigma}_S^2}{\sigma_S^2} \right] \right) \tag{3.11}$$

Furthermore on using the maximum likelihood estimates for the variance terms, we get the log likelihood ratio to be;

$$-\ln \ell = N'_R (\ln \hat{\sigma}_P - \ln \hat{\sigma}_R) + N'_S (\ln \hat{\sigma}_P - \ln \hat{\sigma}_S) \tag{3.12}$$

The log likelihood ratio $-\ln \ell$ is compared with an optimally chosen threshold "*h*" where the threshold was exceeded were considered to be change points. That is,

$$\begin{aligned} -\ln \ell > h &\Longrightarrow H_1 \quad \text{change} \\ -\ln \ell \leq h &\Longrightarrow H_0 \quad \text{no change} \end{aligned} \tag{3.13}$$

The above mentioned method is suitable for independent variables, in other words we can use this method on single variable therefore when we have more then one variable we have multiple results regarding one issue. In order to overcome this drawback of model *Totthan et al* have suggested a combination matrix [4] which is also suggested in [1].

Let us define the hypothesis again, H_0 is implying no change and H_1 is implying change. The expression for *l* is a sufficient statistic and is used to perform a binary

hypothesis test. Under the hypothesis H_0, implying that no change is observed between the two windows H_0 implying no change; likelihood l_o under hypothesis H_0, H_1 implying that a change is observed between the two windows we have, $l_1 = l$. In order to obtain a value for the likelihood ratio η that is bounded between [0 1], we define η as follows;

$$\eta = \frac{l_0}{l_1 + l_0} \tag{3.14}$$

Furthermore, on using the maximum likelihood estimates for the variance terms in equations (3.9) and (3.10) we get;

$$\eta = \frac{\hat{\sigma}_R^{-N'_R} \hat{\sigma}_S^{-N'_S}}{\hat{\sigma}_R^{-N'_R} \hat{\sigma}_S^{-N'_S} + \hat{\sigma}_P^{-(N'_R + N'_S)}} \tag{3.15}$$

and $N'_R = N_R - p \quad N'_S = N_S - p$, p is the order of AR model.

There is a new definition of Likelihood ratio in [32]. *Qingtao et al* have called Log likelihood Ratio (LLR) η_L as follows;

$$\eta_L = \log \frac{l_1}{l_0} \tag{3.16}$$

And after simplification, we get

$$\eta_L = \log \frac{\hat{\sigma}_P^{-(N'_R + N'_S)}}{\hat{\sigma}_R^{-N'_R} \hat{\sigma}_S^{-N'_S}} \tag{3.17}$$

First a $(1 \times n)$ input vector φ is constructed with components of likelihood ration η and $\varphi(t)$ is the Abnormality Vector which defined;

$$\varphi = [\eta_1 .....\eta n] \tag{3.18}$$

The operator matrix A was designed to obtain a scalar value of the measure of the transformation we perform the following operation:

$$\varphi A \varphi^T = \lambda_d \tag{3.19}$$

For example $\varphi(t)$ can chosen like $\varphi_{ip} = \alpha_R [\eta_{IR} . \eta_{IDe} . \eta_{OR.}]$. η_{IR}, η_{IDe}, and $\eta_{OR.}$ represents the MIB variables IPinreceives, IPindelivers, and IPoutrequest respectively. So operator A matrix is represented like;

$$A_{ip} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3.20}$$

Elements of matrix A is composed of spatial correlation between IP variables. Such as, the coupling between ipIDe and ipOR is a_{23} and by symmetry $a_{21}=a_{12}$, $a_{31}=a_{13}$, and $a_{23}=a_{32}$.The operator matrix A has suggested in [4] a static matrix;

$$A_{ip} = \begin{bmatrix} 0{,}87 & 0{,}08 & 0{,}05 \\ 0{,}08 & 0{,}6 & 0{,}32 \\ 0{,}05 & 0{,}32 & 0{,}63 \end{bmatrix} \tag{3.21}$$

There was a different operator A matrix definition in [1] which is based on abnormality vector $\varphi(t)$ and we call this matrix $A2$ type Operator Matrix in Results. A_{ip} defined as;

$$A_{ip} = \left| \langle \varphi_i(t), \varphi_j(t) \rangle \right| \tag{3.22}$$

$$A_{ip} = \frac{1}{T} \left| \sum_{t=1}^{T} \varphi_i(t) \varphi_j(t) \right| \tag{3.23}$$

This is ensemble average of the two point spatial cross-correlation of the abnormality vectors over a time interval T [1]. We can define λ_d same as (3.19).

All the above mentioned method is written based on [1, 3, 4]. From now on we will add our contribution to this method. The above mentioned method is based on GLR(Generalized Likelihood Ration) test and the AR estimation part is based on Least Squares Estimation [5].

Our contribution is in three parts; First part is using Yule-Walker Estimate instead of LSE. Second part is using Modified Yule-Walker estimation. Third part is using SVD (Singular Value Decomposition) to define *A* operator matrix. These contributions are design by me (Umut Güven) and my two friends. They are Dağhan Hasan and Derya Erhan. These contributions are a product of a collaborative study of us. We will briefly explain our work and we will conclude with my contribution to these methods which is Wavelet-AR model and we will give some test results in order to show our contribution to above mentioned method.

3.2. Yule Walker Method

We will use the Yule Walker Methods to estimate the AR parameters. We will try to estimate variance terms σ_R^2, σ_S^2, and σ_P^2 these are learning window variance, test window variance, and pooled window variance respectively. *Y* is the time series vector and it is representing one of the MIB variables. A time window of size *N (N=10)* and *p* is the AR model order. Let's define covariance $r(i)$;

$$r(i) = \frac{1}{N}\sum_{i=0}^{p}\{[y(1), y(2),...., y(N-i)]*[y(i+1),...y(N)]\} \tag{3.24}$$

When we acquire $r(i)$ next step is the estimating the AR parameters;

$$\begin{bmatrix} r(0) & r(-1) & \cdots & r(-p) \\ r(1) & r(0) & \cdots & \vdots \\ \vdots & \vdots & \ddots & r(-1) \\ r(p) & \cdots & \cdots & r(0) \end{bmatrix} * \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.25}$$

The above equations (3.24, 3.25) are called the Yule-Walker equations or Normal equations. We could write the equation (3.25) in different way which gives us clear results for AR parameters.

$$\theta = [a_1, \ldots\ldots, a_p]^T \tag{3.26}$$

$$\begin{bmatrix} r(1) \\ \vdots \\ r(p) \end{bmatrix} + \begin{bmatrix} r(0) & \cdots & r(-p+1) \\ \vdots & \ddots & \vdots \\ r(p-1) & \cdots & r(0) \end{bmatrix} * \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.27}$$

With clear definition,

$$r_p + R_p \theta = 0 \tag{3.28}$$

The solution is $\theta = -R_p^{-1} r_p$. Once θ is found, σ^2 can be found from the first row of Equation (3.25). The Yule-Walker method for AR spectral estimation is based directly on (3.25). Given data $y(t)$, we first obtain sample covariance $r(i)$, define the equation (3.25) and solve the equation (3.28). When we found the θ we can easily found the variance (σ^2) of given vector [6].

3.3. Modified Yule-Walker Method

The modified Yule-Walker is a two stage procedure for estimating the ARMA (Autoregressive Moving Average) spectral density. In the first stage we estimate the AR coefficients; second stage is estimating the MA (Moving Average) part of ARMA spectrum [6]. In this work we only focus on the AR parameter estimation. Let's define the parameters; *Y*: the data vector, *p*: AR model order, m: MA model order, *M*: the constant which determine the amount of overdetermination. We accept $m = 0$, and we continue the model definition.

$$\begin{bmatrix} r(0) & r(-1) & \cdots & r(-p+1) \\ r(1) & r(0) & \cdots & r(-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ r(M-1) & \cdots & \cdots & r(-p+M) \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(M) \end{bmatrix} \tag{3.29}$$

If we set $M = p$ in (3.29) we obtain a system of p equations in p unknowns. This represents a generalization of the Yule-Walker equations that holds in the AR case. Replacing the theoretical covariance $r(k)$ by their sample estimates $\hat{r}(k)$ in (3.29), equation (3.29) became Modified Yule-Walker system equation and $a_i's$ became $\hat{a}_i$ modified Yule-Walker estimates.

We can exploit the additional information by choosing $M > p$ in equation (3.29) and solving the overdetermined system of equations. A most common way to overcome this problem is to solve the resultant equation;

$$\hat{R}\hat{a} \cong -\hat{r} \tag{3.30}$$

In a least squares (LS) sense. For instance, the least squares solution to (3.30) is given by;

$$\tilde{a} = -(\hat{R} * W\hat{R})^{-1}(\hat{R}^{*}W\hat{r}) \tag{3.31}$$

Where W is an MxM positive definite weighting matrix. Choosing $M > p$ does not always improve the accuracy of above mentioned AR coefficient estimates. In fact, if the poles and zeros are not close to unit circle, choosing $M > p$ can make the accuracy worse. A simplified first choice is $W = I$, resulting in the regular least squares estimate. Most accuracy improvement can be realized by choosing $M > p$ and $W = I$ for many problems [6].

Using the previous calculated estimates of a_k and r_k we can define following estimator of γ_k; we have previously stated that we use $m = 0$,

$$\hat{\gamma}_k = \left\{ \sum_{j=0}^{n} \sum_{p=0}^{n} \hat{a}_j \hat{a}_p * \hat{r}(k+p-j) \right.$$
$$k = 0,\ldots,m \tag{3.32}$$
$$\hat{a}_0 = 1$$

$\hat{\gamma}_k$ Term gives us the variance estimate of the Modified Yule-Walker method.

3.4. SVD to Define A Operator Matrix

We use the SVD (Singular value decomposition) to define a dynamic *A* operator matrix. In general case SVD define as; suppose *S* is an *d*-by-*e* matrix whose entries come from the field *K*, which is either the field of real numbers or the field of complex numbers. Then there exists a factorization of the form.

$$S = U \sum V^* \tag{3.33}$$

Where *U* is a *d*-by-*d* unitary matrix over *K*, the matrix *Σ* is *d*-by-*e* with nonnegative numbers on the diagonal and zeros off the diagonal, and V^* denotes the conjugate transpose of *V*, an *e*-by-*e* unitary matrix over *K*. Such a factorization is called a singular-value decomposition of *S* [7]. The matrix *V* thus contains a set of orthonormal "input" or "analyzing" basis vector directions for *S* The matrix *U* contains a set of orthonormal "output" basis vector directions for *S*. The matrix *Σ* contains the singular values, which can be thought of as scalar "gain controls" by which each corresponding input is multiplied to give a corresponding output.

When we turn back to Wavelet-AR or AR models, we have stated that we have abnormality vector φ in Equation (3.18) and our decision variable λ_d is;

$$\lambda_d = \varphi^H A \varphi \tag{3.34}$$

We want to create an A operator matrix based on property of orthonormality of Singular Value Decomposition (SVD) method. We can calculate R_φ covariance matrix;

$$R_\varphi = E\left[\varphi\ \varphi^H\right] \tag{3.35}$$

When we look at the correlation of variables with (3.35) we have seen high correlation therefore we want to decorrelate the variables in order to give weight them, and we have applied eigen decomposition to R_φ covariance matrix. Since we have done this decomposition with Matlab's SVD command we have named this decomposition in whole thesis as a "SVD" method. We can estimate R_φ with equation (3.36).

$$\hat{R}_\varphi = \frac{1}{N}\sum_i \varphi_i \varphi_i^H \tag{3.36}$$

This can be written as by making singular value decomposition:

$$\hat{R}_\varphi = U \sum U^H \tag{3.37}$$

By doing a whitening transform or decoration;

$$\underline{v} = U^H \varphi \tag{3.38}$$

$$E[\underline{v}\underline{v}^H] = U^H R_\varphi U = \sum \tag{3.39}$$

Creating A operator Matrix;

$$\underline{v} = U^H \varphi \ \text{ and also } \ \varphi = U\underline{v} \tag{3.40}$$

Multiplication of unitary matrixes must be identity matrix.

$$UU^H = U^H U = I \tag{3.41}$$

If we change φ by $\varphi = U\underline{v}$ in (3.34) we get;

$$\lambda_d = \underline{v}^H (U^H A U)\underline{v} \tag{3.42}$$

In equation (3.42) $\underline{v}$ and $\underline{v}^H$ are decorrelated items. If we say;

$$\lambda_d = \underline{v}^H B \underline{v} \tag{3.43}$$

B Matrix should be diagonal and we do not want to multiply decorarelated items.

$$B = \begin{bmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{bmatrix} \tag{3.44}$$

$$B = U^H A U \tag{3.45}$$

$$A = U B U^H \tag{3.46}$$

The v values can be chosen like $v_1 = 1/\sigma_1^2$ and similarly for the other values with their corresponding variances. These σ_i^2 correspond to $\sum_{ii}$ of components of $\underline{v}$. If we want, we can also choose these σ_i^2 arbitrary in order to give weight one of them. Then *A* operator matrix became,

$$A = U \begin{bmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{bmatrix} U^H \tag{3.47}$$

We have define the A operator matrix based on property of orthonormalilty of Singular Value Decomposition (SVD) method. This matrix is can be created *dxd* independent of the number of input vectors this gives us a flexibility to work with different number of input vectors.

3.5. Correlation Coefficient Method to Create A Operator Matrix

There is a "corrcoef" function in Matlab 7.0, R14, this function calculates the correlation coefficients based on given input matrix *X*. in general form Correlation Coefficients are defined as: In probability theory and statistics, correlation, also called correlation coefficient, indicates the strength and direction of a linear relationship between two random variables. In general statistical usage, correlation or co-relation refers to the departure of two variables from independence, although correlation does not imply causation. In this broad sense there are several coefficients, measuring the degree of correlation, adapted to the nature of data [8].

In Matlab 7.0, R14, there is a "corrcoef" function and its definition is;
R = corrcoef(*X*); *R* = corrcoef(*X*) returns a matrix *R* of correlation coefficients calculated from an input matrix *X* whose rows are observations and whose columns are variables. The matrix *R* = corrcoef(*X*) is related to the covariance matrix *C* = cov(*X*) by

$$R(i,j) = \frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}} \tag{3.48}$$

We can use above mentioned *R* matrix instead of *A* operator matrix. All the above mentioned methods easily explain by Figure 3.2.

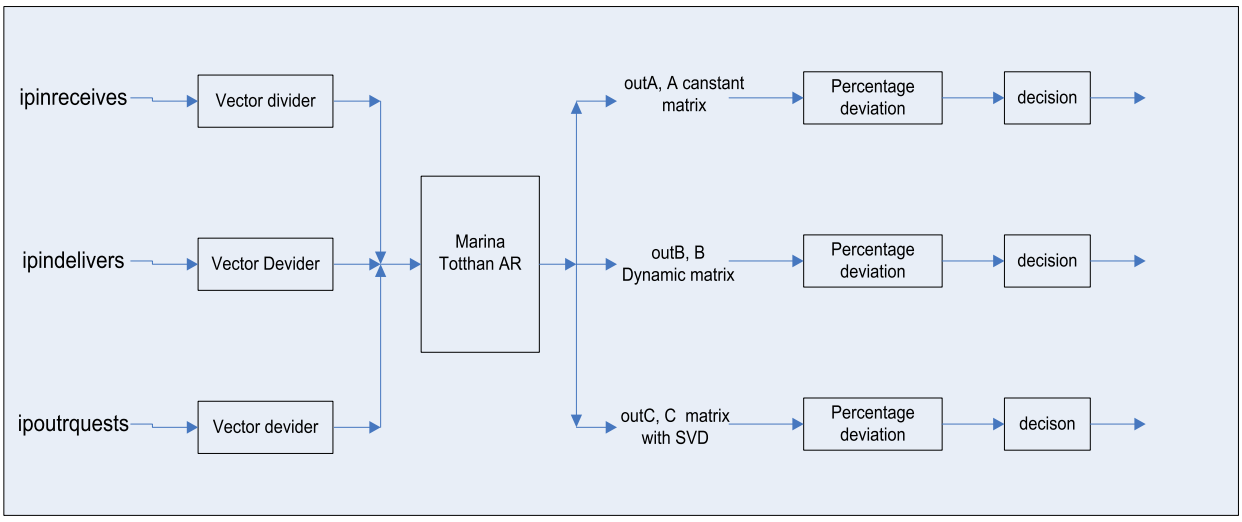

Figure 3.2. AR models

As you can see from Figure 3.2. we have used Percentage Deviation algorithm for detection phases.

3.6. Percentage Deviation

Percentage deviation algorithm is based on median calculation of input vector of $\underline{x}$,

$$PD_X = (x - medain(\underline{x})) * 100 \tag{3.49}$$

In probability theory and statistics, a median is a number dividing the higher half of a sample, a population, or a probability distribution from the lower half. The median of a finite list of numbers can be found by arranging all the observations from lowest value to highest value and picking the middle one [9]. Then we will go one step further, average of Percentage Deviation;

$$PD_{avg} = \sum_{i=1}^{n} PD_i / n \tag{3.50}$$

This PD_{avg} gives us the decision threshold. If $PD_x > PD_{avg}$ it shows us there is an anomaly at point $x(i)$, else is normal condition.

4. METHODOLOGY - WAVELET

4.1. Wavelet- Modulus Maxima Model

We have used Modulus Maxima function for abnormality analysis via Wavelet and we have applied the Percentage Deviation for detection. We will try to explain briefly in this section and we want to show the performance of two wavelet models also. Figure 4.1. is explaining our Wavelet model, we will mention about Modulus Maxima function.

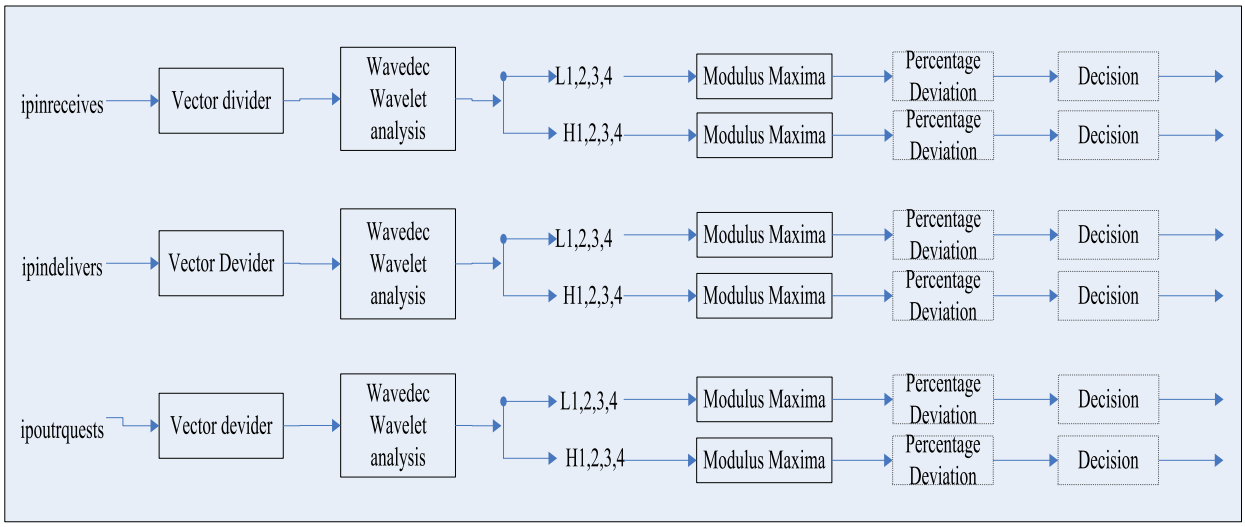

Figure 4.1. Wavelet model

For edge or singularity detection, we are only interested in the local maxima of $\left|W^1(\lambda,t)\right|$. When detecting the local maxima of $\left|W^1(\lambda,t)\right|$, we can also keep the value of the wavelet transform at the corresponding location [13].

Let us briefly define what we mean by local maxima of the wavelet transform modulus. Let $W(\lambda,t)$ be wavelet transform of a function $x(t)$.

- We call local extremum any point (λ_0,t_0) such that $(\partial W(\lambda_0,t))/(\partial(t))$ has zero crossing at $t=t_0$, when t varies.

- We call modulus maximum, any point (λ_0, t_0) such that $|W(\lambda_0, t)| < |W(\lambda_0, t_0)|$ when t belongs to either a right or the left neighborhood of t_0, and $|W(\lambda_0, t)| \leq |W(\lambda_0, t_0)|$ when t belongs to the other side of the neighborhood of t_0.

- We call maxima line, any connected curve in the scale pace (λ, t) along which all points are modulus maxima.

A modulus maximum (λ_0, t_0) of the wavelet transform is a strict local maximum of the modulus either on the right or the left side of the t_0 [13].

4.2. Wavelet-AR Model

We try to combine Wavelet Analysis and AR model in order to detect intrusions. And our implementation results eager us to study this model more deeply. Wavelet Analysis will be explained in this section and we will try to explain our Wavelet-AR model.

Wavelets are mathematical tools for analyzing time series or images. Our work on wavelet in this study focuses on their use with time series. We focus on discrete time series analysis. Wavelets are a relatively new way of analyzing time series in that the formal subject dates back to the 1980s, but in many aspects wavelets are a synthesis of older ideas with new elegant mathematical results and efficient computational algorithms [11].

Wavelet is a "small wave". A small wave grows and decays essentially in limited time period. In order to quantify the notation of a wavelet, let us consider a real-valued function $\psi(\cdot)$ defined over the real axis $(-\infty, \infty)$ and satisfying two basic properties

1. The integral of $\psi(\cdot)$ is zero:

$$\int_{-\infty}^{\infty} \psi(u)du = 0 \tag{4.1}$$

2. The square of $\psi(\cdot)$ integrates to unity.

$$\int_{-\infty}^{\infty} \psi^2(u)du = 1 \tag{4.2}$$

Hence equations (4.1) and (4.2) lead to “small wave” or wavelet [11]. Two different wavelets are plotted in Figure 4.2.

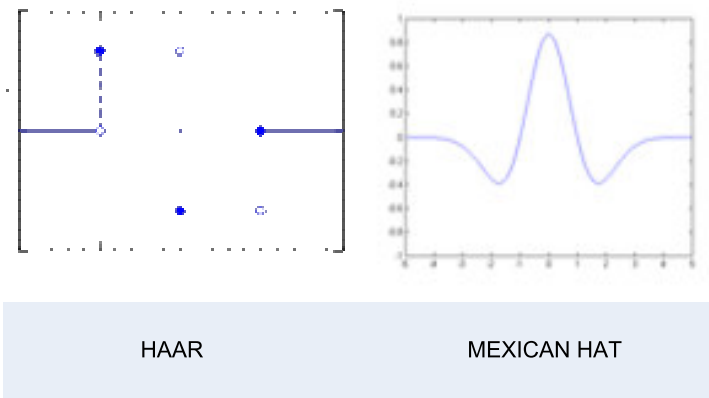

Figure 4.2. Haar and Mexican Hat wavelets

The Haar wavelet is one of the oldest wavelet. The first wavelet called the Haar wavelet and its function;

$$\psi^{(H)}(u) \equiv \begin{cases} -1/\sqrt{2}, & -1 < u \le 0; \\ 1/\sqrt{2}, & 0 < u \le 1 \\ 0, & otherwise \end{cases} \tag{4.3}$$

Wavelets can tell us how weighted averages of certain other functions vary from one averaging period to the next. This interpretation of wavelet analysis is a key concept. Let us explain briefly. Let $x(\cdot)$ real-valued function, we can define $\alpha(a,b)$ average value of $x(\cdot)$

$$\frac{1}{b-a}\int_a^b x(u)du \equiv \alpha(a,b), \tag{4.4}$$

We can easily consider it to be a function of the length of the interval $\lambda \equiv b-a$ center of interval $t=(a+b)/2$. We refer to λ scale associated with average. Using λ and t, we can define,

$$A(\lambda,t) \equiv \alpha(t-\frac{\lambda}{2},t+\frac{\lambda}{2}) = \frac{1}{\lambda}\int_{t-\frac{\lambda}{2}}^{t+\frac{\lambda}{2}} x(u)du. \tag{4.5}$$

We call $A(\lambda,t)$ the average value of the signal $x(\cdot)$ over a scale of λ centered about time t. If we want to measure changes between to averages we can define;

$$D(\lambda,t) \equiv A(\lambda,t+\frac{\lambda}{2}) - A(\lambda,t-\frac{\lambda}{2}) = \frac{1}{\lambda}\int_{t}^{t+\lambda} x(u)du - \frac{1}{\lambda}\int_{t-\lambda}^{t} x(u)d(u). \tag{4.6}$$

For example, A plot of $D(1,t)$ would tell us, how quickly the daily average temperature is changing from one day to the next. Similarly, by increasing the scale λ up to a year, a plot of $D(1,t)$ would tell us how much the yearly average temperature is changing from one year to next. Now we can combine these explanations to wavelets. Because two integral in Equation (4.6) involve adjacent nonoverlapping intervals, it is easy to combine them into a single interval over the entire real axis to obtain;

$$D(\lambda,t) = \int_{-\infty}^{+\infty} \tilde{\psi}_{\lambda,t}(u)x(u)du, \tag{4.7}$$

Where,

$$\tilde{\psi}_{\lambda,t}(u) \equiv \begin{cases} -1/\lambda, & t-\lambda < u \le t; \\ 1/\lambda, & t < u \le t+\lambda; \\ 0, & otherwise \end{cases} \tag{4.8}$$

If we compare Equation (4.8) to Haar wavelet we see that $\tilde{\psi}_{1,0}(u) = \sqrt{2}\psi^{(H)}(u)$. the scheme of looking at differences on unit scale, to integrating the product of the signal $x(\cdot)$;

$$\int_{-\infty}^{\infty} \psi^{(H)}(u)x(u)du \equiv W^{(H)}(1,0), \tag{4.9}$$

The Haar wavelet extract information about how much difference there is between the two unit scale averages of $x(\cdot)$ bordering on time $t = 0$ [11].

We have define the wavelet analysis based on continuous time signal we will briefly explain the DWT (Discrete Wavelet Transform). DWT can be thought of as a judicious subsampling of $W(\lambda, t)$ in which we deal with just dyadic scales and then within a given dyadic scale 2^{j-1}, peak times t that are separated by multiples of 2^j [11].

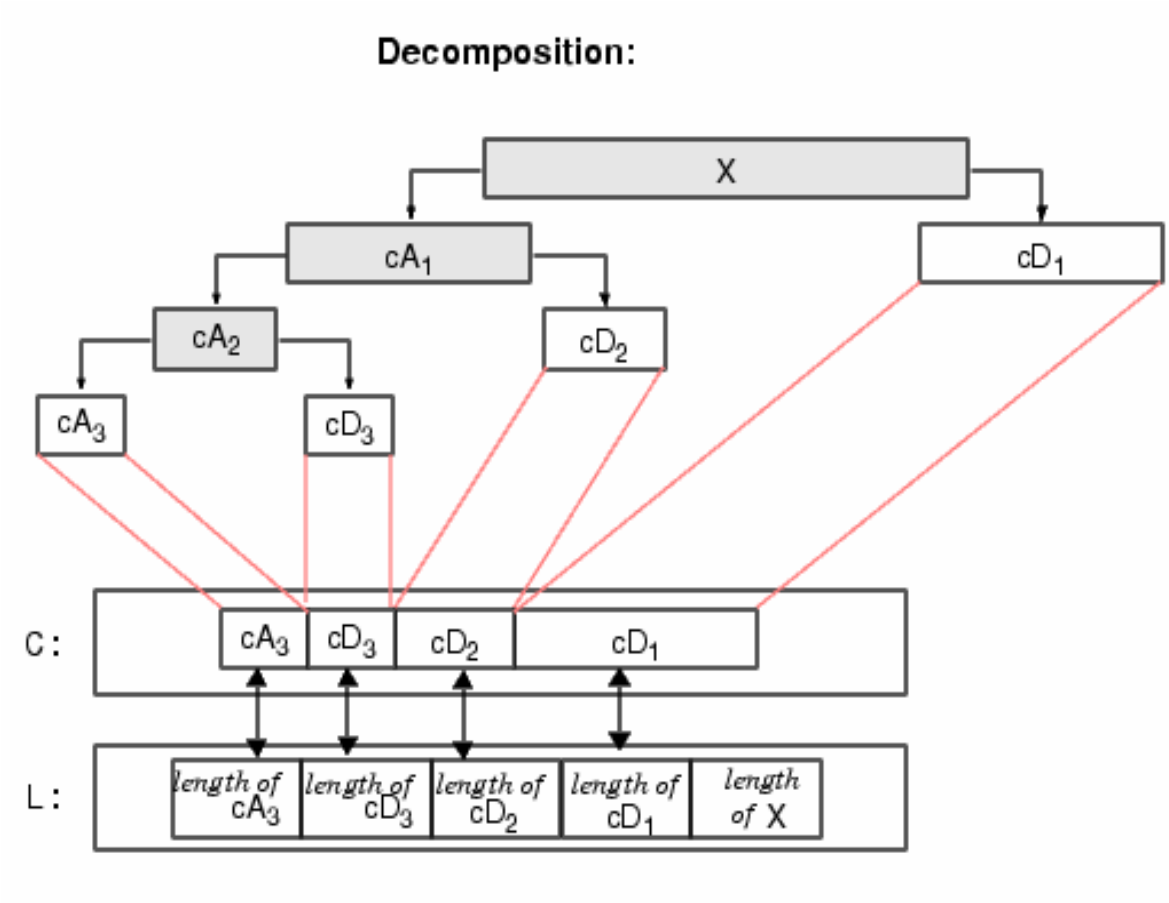

Figure 4.3. DTW follow cart

We will go one step forward and we will briefly explain the MODWT (Maximal Overlap DWT), MODWT can be thought of as a subsampling of the CWT (Continuous

Wavelet Transform) at dyadic scales, but, in contrast to DWT, we now deal with all times t and not just that are multiples of 2^j [11].

We will briefly explain the detailed coefficients, approximate coefficients, Wavelet filter, Scaling filter, Low Pass filter, High Pass filter and relation between these phenomenons.

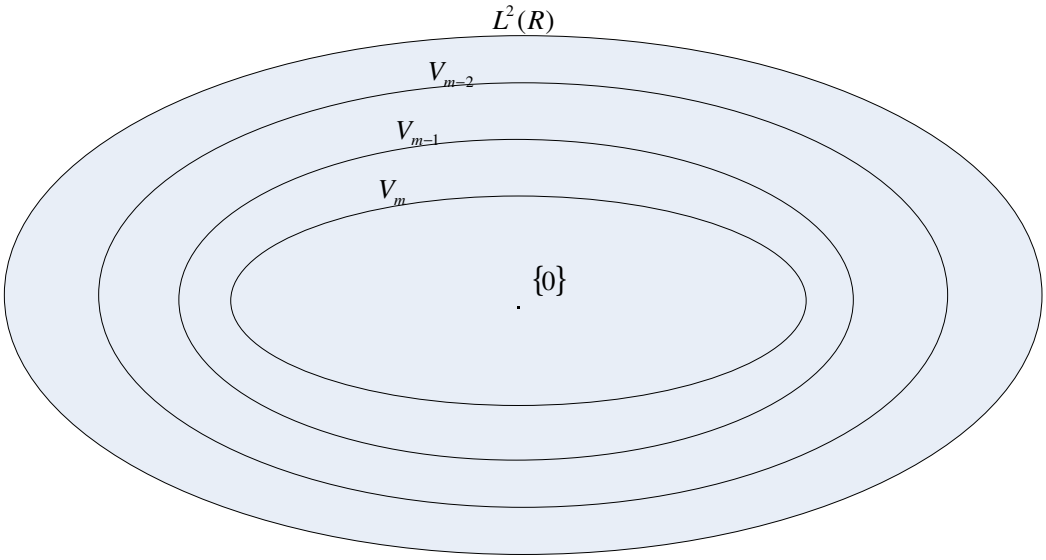

Figure 4.4. Multiresolution representation of $L^2(R)$

The wavelet representation of functions in $L^2(R)$, when observing the graph of the multiresolution representation (Figure 4.4.), we see that the space $L^2(R)$ is built up of the set of "rings" that are differences between two consecutive spaces. These difference spaces are denoted by W_m and are defined as orthogonal complement of spaces V_m with respect to V_{m-1},

$$V_{m-1} = V_m \oplus W_m, \quad V_m \perp W_m \tag{4.10}$$

Let $\psi(x) = \psi_{0,0}(x)$ be a basis function of W_0. Since $\psi_{0,0}(x) \in W_0 \subset V_{-1}$ we can write;

$$\psi_{0,0}(x) = 2^{1/2} \sum_n g_n \phi_{-1,n}(x), \tag{4.11}$$

$\phi(x), g_n$ are scaling filter and high pass filter respectively. There are strong relation between $\psi(x)$, $\phi(x)$, g_n, and h_n The first;

$$\Phi(\omega) = \prod_{m=1}^{\infty} H(\frac{\omega}{2^m}) \tag{4.12}$$

Rewriting (4.11) in frequency domain,

$$\Psi(\omega) = G(\frac{\omega}{2})\Phi(\frac{\omega}{2}) \tag{4.13}$$

And replacing $\Phi(\omega)$ using the infinite product of (4.12) yields,

$$\Psi(\omega) = G(\frac{\omega}{2})\prod_{m=1}^{\infty} H(\frac{\omega}{2^m}) \tag{4.14}$$

Next, we define the relation between two sequences g_n and h_n. We can state that;

$$g_n = (-1)^n h_{-n+2t+1} \tag{4.15}$$

The equivalent of (4.15) in frequency domain is;

$$G(\omega) = -H(-\omega + \pi)e^{+i\omega(2t+1)} \tag{4.16}$$

The introduction of the wavelet functions enables us to write any function $x(\cdot)$ in $L^2(R)$ as a sum of projections on W_j, $j \in R$,

$$x(t) = \sum_{j=-\infty}^{\infty} e_j(t) \tag{4.17}$$

Where;

$$e_j(t) = \sum_k \langle \psi_{j,k}(t), x(t) \rangle \psi_{j,k}(t) \tag{4.18}$$

If we stop at a certain scale *m*, then the function $x(\cdot)$ can be written as the sum of a low resolution part $x_m(t) \in V_m$ and number of detailed parts $e_j(t) \in W_j$;

$$\begin{aligned} x(t) &= x_m(t) + \sum_{j=-\infty}^{m} e_j(t) \\ &= \sum_n \langle \phi_{m,n}(t), x(t) \rangle \phi_{m,n}(t) + \sum_{j=-\infty}^{m} \sum_k \langle \psi_{j,k}(t), x(t) \rangle \psi_{j,k}(t) \end{aligned} \tag{4.19}$$

$$x(t) = \sum_n c_{m,n} \phi_{m,n}(t) + \sum_{j=-\infty}^{m} \sum_k d_{j,k} \psi_{j,k}(t) \tag{4.20}$$

We want to define the DWT, let $x(n)$ is the discrete version of it's continuous time version and can be decomposed in two functions $x_1(n) \in V_1$ and $e_1(n) \in W_1$ containing the overall characteristics and details of $x_0(n)$, respectively,

$$x_0(n) = x_1(n) + e_1(n) = \sum_k c_{1,k} \phi_{1,k}(x) + \sum_k d_{1,k} \psi_{1,k}(x) \tag{4.21}$$

Now two new sequences $c_{1,k}$ and $d_{1,k}$ have been generated. $c_{1,k}$ is the approximate coefficients and $d_{1,k}$ is the detailed coefficients. We have defined the DWT, and it is possible to calculate iteratively coefficients $c_{1,k}$ and $d_{1,k}$ from the previous scale $j-1$ without explicit use of the functions $\phi(x)$ and $\psi(x)$. We can define for arbitrary j ,

$$c_{j,k} = 2^{1/2} \sum_n c_{j-1,n} h_{n+2k,} \tag{4.22}$$

$$d_{j.k} = 2^{1/2} \sum_n c_{j-1,n} g_{n+2k,} \tag{4.23}$$

Description of the decomposition process is completely discrete. The sequences h_n and g_n are called filters.

Now we can define our Wavelet-AR model based on all above mentioned theoretical explanations. You can see the Figure 4.5.

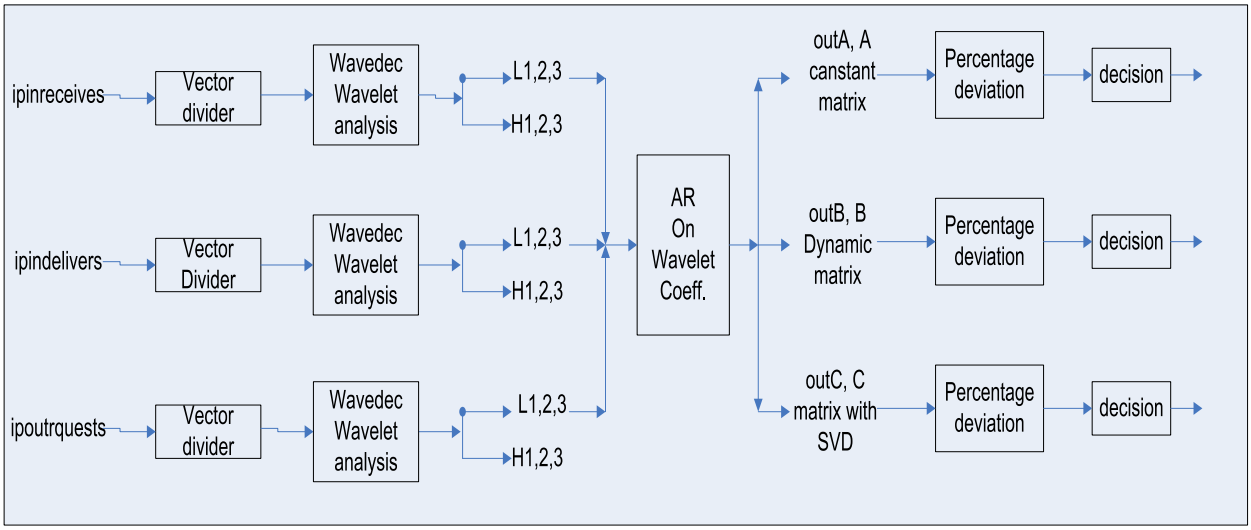

Figure 4.5. Wavelet-AR model

We will explain briefly the functions of Wavelet-AR model step by step. The "Vector Divider" function is only a Matlab implementation function. It divide the vectors and then we can perform the analysis on divided vectors using less CPU power, it decreases the computational burden of model. At the end we combined the divided vectors to one vector its length is almost equal to input vector.

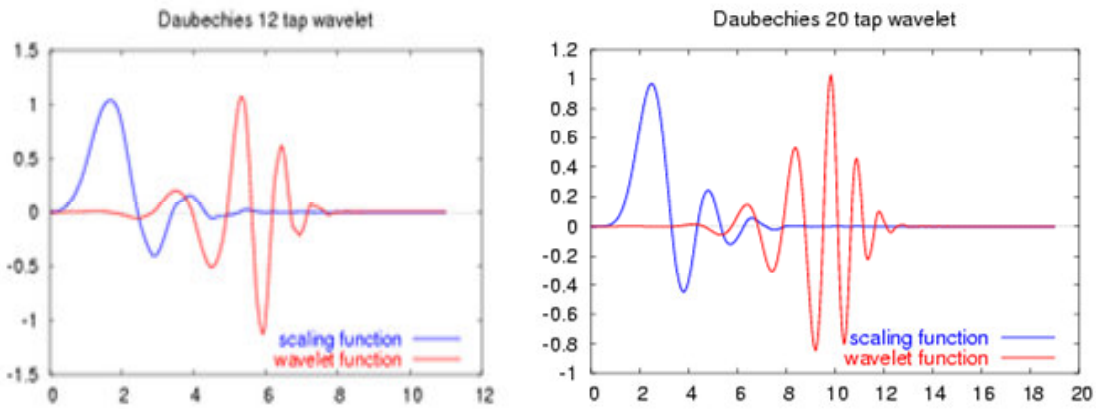

Figure 4.6. Daubechies wavelets and their scaling filters

The “Wavelet Analysis” function is a Matlab function and defined as “Wavedec” [*C, L*] = wavedec(*X*, *N*, 'wname') returns the wavelet decomposition of the signal *X* at level *N*, using 'wname'. *N* must be a strictly positive integer. We can choose pre-defined Mother-Wavelets in Matlab, Figure 4.6. shows Daubechies Wavelet and its Scaling filter. The output decomposition structure contains the wavelet decomposition vector *C* and the bookkeeping vector *L* [12].

We use only the low pass filter out of Wavelet analysis; moreover we use only the approximated coefficients $c_{1,k}$ of Wavelet analysis. We feed approximate coefficients to AR function as an input vector. We do this in two different ways;

First one is, we combine all the approximate coefficients of all three inputs and when the level is three this combination gives us 9x9 A operator matrix and nine abnormality vectors correspond to all input vector’s all levels. If we use level two for Wavelet analysis, combination gives us 6x6 A operator matrix and six abnormality vectors. If we use level one Wavelet analysis combination gives us 3x3 or if we use LAN variable it gives us 4x4 A operator matrix and four abnormality vectors. Figure 4.7. sample *A* operator matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 4.7. A operator matrixes 6x6, 4x4, 3x3

Second one is, we perform AR analysis level by level. If we perform Wavelet analysis in three levels it gives us three low pass coefficient for each input vector. Than we perform AR analysis on wavelet coefficients level by level. If we perform level three for Wavelet analysis it gives us three low pass wavelet coefficients and it creates a 3x3 *A* operator matrix for each level and abnormality vectors for each level. If we use level one Wavelet analysis it gives us one approximate Wavelet coefficient $c_{1,k}$ for each input vector

then we perform AR analysis on each approximate coefficients it gives us 3x3 one *A* operator matrix, and three abnormality vectors.

The idea behind the using wavelet analysis is eliminating the high variations from input vector while do not affecting the input vector's originality. These high variations in input vector can prompt the AR model to create false alarms because of the AR model is very sensitive to changes. It gives us good detection and beside this it gives us high False Positive Rate. We will explain more deeply the results on results section with multiple examples. Next section we will show the performance comparison all above mentioned methods.

4.3. Wavelet-AR Model HW Considerations

The size of the implementation of an algorithm depends strongly on the minimum feature size of the technology. It also depends on the specific circuit design style, such as CMOS or DCVSL, and the number of available metal layers for wire routing. Hence, it is necessary to resort to an approximate, technology and circuit style independent measure. A commonly used measure for the size of a design is the number of NAND gate equivalents (GE). This is the area of the circuit implementation divided by the area of the smallest NAND gate in the used standard CMOS cell library. Table 4.1. below contains a subset of logical gates taken from a standard cell library for 130 nm CMOS technology. [34].

Table 4.1. Hardware costs of logical operations

Logical operation	Binary function	Hardware cost
NAND(a, b)	$ab + 1$	1.00 GE
NOR(a, b)	$1 + a + b + ab$	1.00 GE
AND(a, b)	ab	1.25 GE
OR(a, b)	$a + b + ab$	1.25 GE
XOR(a, b)	$a + b$	2.25 GE
NAND(a, b, c)	$abc + 1$	1.25 GE
NOR(a, b, c)	$1 + a + b + c + ab + ac + bc + abc$	1.50 GE
AND(a, b, c)	abc	1.50 GE
OR(a, b, c)	$a + b + c + ab + ac + bc + abc$	1.75 GE
XOR(a, b, c)	$a + b + c$	4.00 GE
MAJ(a, b, c)	$ab + ac + bc$	2.25 GE
MUX$(a, b; c)$	$a + ac + bc$	2.50 GE

Using this information we will try to determine the hardware equivalent of the Wavelet-AR Intrusion Detection System.

Table 4.2. Gate Equivalent of Wavelet-AR

Operations		Number of Operation	Gate Equivalent
Wavelet Analysis	IPinreceives	15000	16875 GE
	IPindelivers	15000	16875 GE
	IPoutrequests	15000	16875 GE
	If_inoctec	15000	16875 GE
Likelihood Ratio	IPinreceives	11900	13388 GE
	IPindelivers	11900	13388 GE
	IPoutrequests	11900	13388 GE
	If_inoctec	11900	13388 GE
A Operator Matrix	SVD Type	7820	8798 GE
Decision	Percentage Deviation	650	732 GE
Wavelet-AR	500 steps data	116070	130582 GE

With these operations we get total 116070 operations and 130582 Gate Equivalent. If we compare these results with a standard PC CPU we see that Wavelet-AR IDS can implement in Real Time. A standard P4 processor Laptop can do 3200 MFLOP (Millions of floating operations per second). We can increase our step size to 1000 step size. At that time 232.140 operations have to be done to reach the decision. When step size is 1000 we also fairly say that number of operations is fewer than P4 CPU operation power so we can implement the Wavelet-AR IDS in real time.

We have taken 1000 samples with one second sample window at 1Mbit. In this case 1000 sample equals to 1000 second and we are doing 232.140 FLOP operations. Number of FLOP operations is very smaller than number of P4 processor FLOP operations.

If we take samples with 1Gbitps and take 1000.000 samples with 1000 samples per second, 1000.000 samples equals to 1000 second and we are doing 232.140 Kilo FLOP operations. In this case number of FLOP operations is 13.78 times smaller than P4 processor FLOP operations. We can reduce the window size to 100 second and we can increase the data rate up to 10Gbitps. Wavelet-AR model can work real-time with 10Gbitps data rate and 100 second observation window with P4 standard Laptop.

5. RESULTS-AR and DATASETS

5.1. DataSet-1

Initially there are only two hosts working in the network, an attacker machine which is used to make DoS attacks by some tools working in Linux platform, and a target machine. The test environment can be seen from Figure 5.1. As seen from Figure 5.2. there is a peak point between sample 104 and 130 in both three MIB variables. This peak is a result of a UDP Flood attack directed to the switch which is our source of SNMP data. There are more attacks that can not be seen directly from the figures of MIB data but can be detected by using some statistical methods. In this work we try to find these statistical methods to find out the DoS attacks that cannot be seen from SNMP data directly. In first dataset there are three DoS attacks, and can be used to test the models detection rate.

As a secondary work there are three hosts working in the network, which are using network in legal ways Figure 5.3. There is no attack present in the network while second dataset was being taken. Since the second dataset have only normal traffic values, it can be used to test false alarm rate of the model. In the following sections we will see the result of the applied methods by using these datasets.

5.1.1. Simulation

In order to test the method there should be SNMP data obtained from a real network. The network topology for collecting SNMP data is shown in following Figure 5.1.

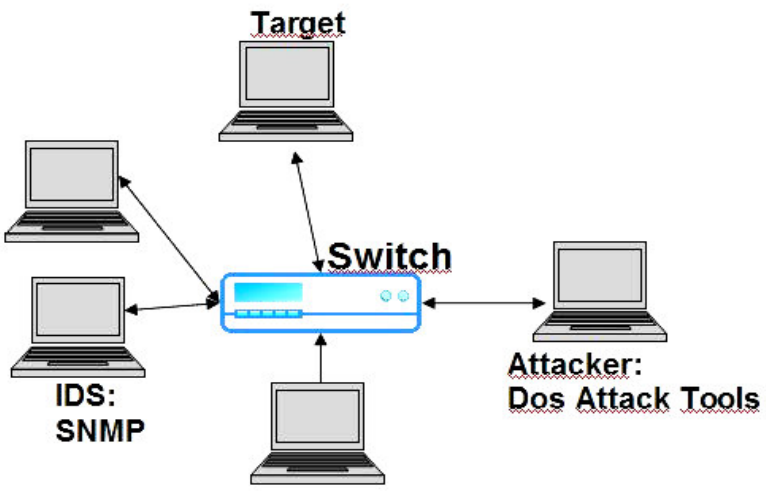

Figure 5.1. Topology for collecting SNMP data. Attacker PC and target PC are in the same network. Attacker PC can be any machine that has IP connectivity with target PC. Linux platform is more useful than windows to use as an attacker PC.

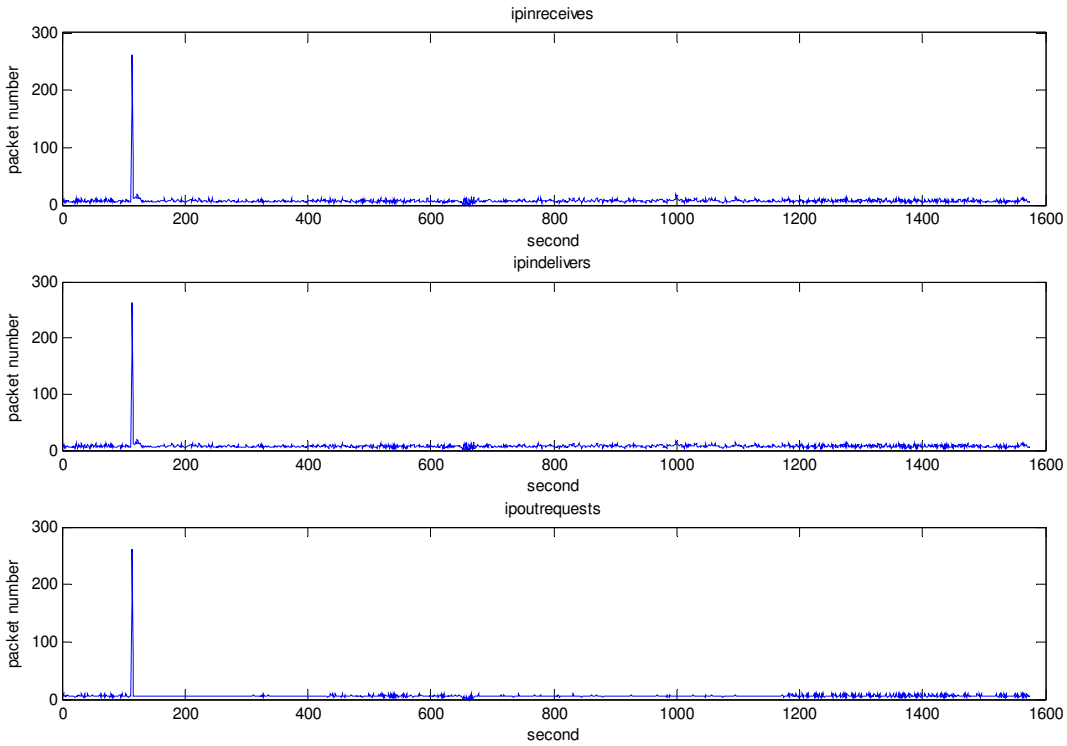

Figure 5.2. First dataset of MIB variables were collected from test network; data with attacks

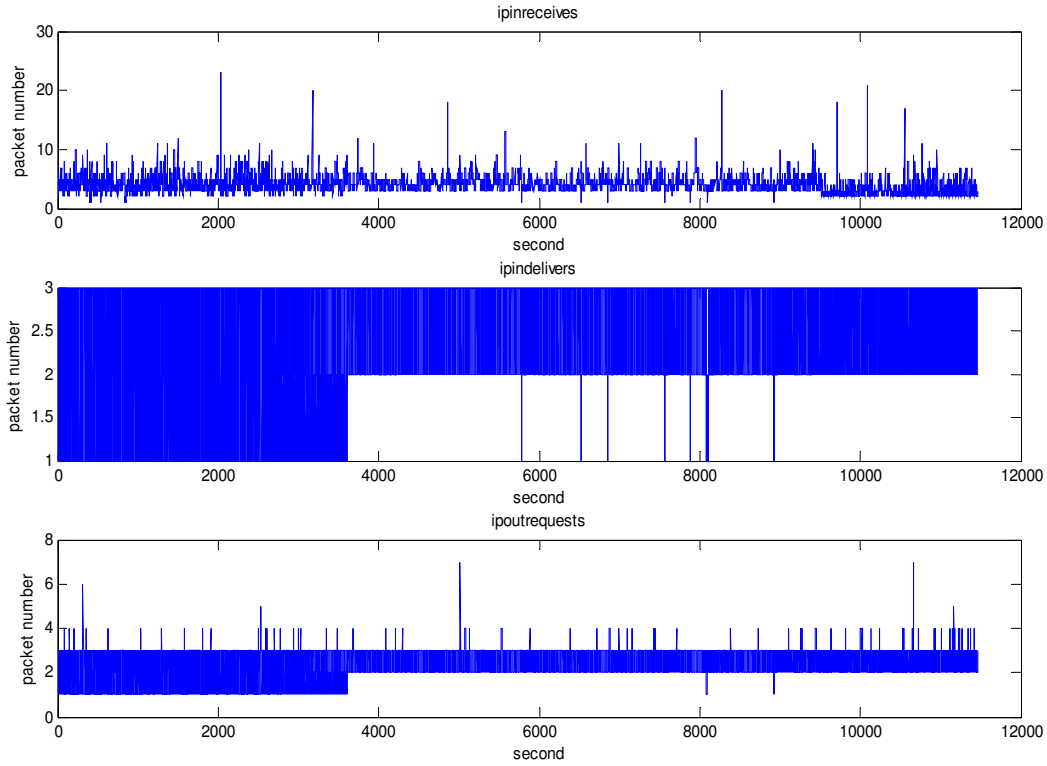

Figure 5.3. Second dataset of MIB variables were collected from test network, attack free data-1

We will mention these datasets; data with attacks and attack free data-1 respectively.

5.2. *Totthan et al* AR(1) LS Estimate

Our first model is the AR model based on Least Squares (LS) Estimation. We use Matlab codes written by R. Moses which has the name "lsar". Than we have applied the Totthan's AR model on it. Figure 5.4. and Figure 5.5. are the results of data with attacks and attack free data-1 respectively. We will add Matlab codes to the Appendix. AR order: one was used in these analyses.

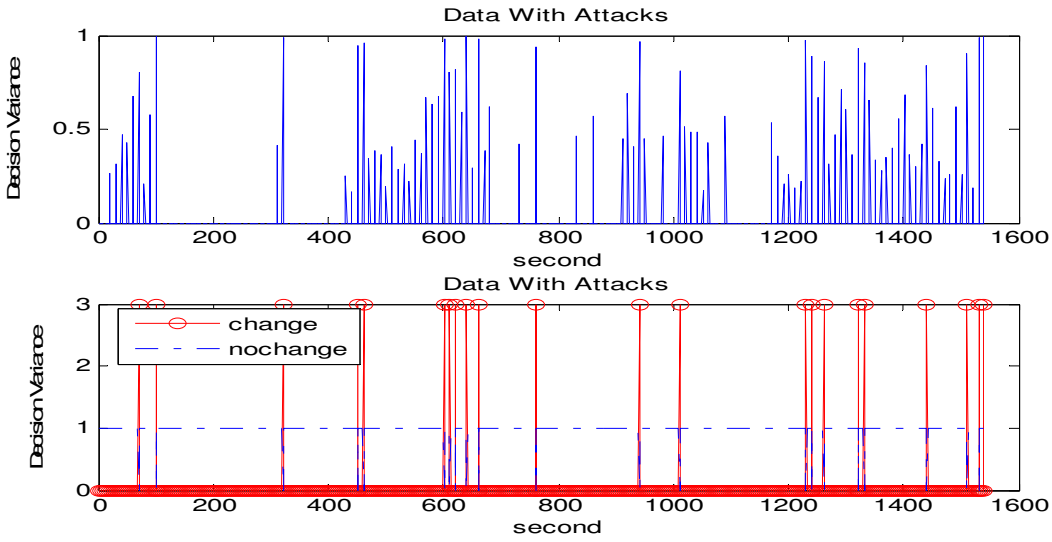

Figure 5.4. Decision variance of data with attacks

There are more than three high decision variances in this plot and because of this there are lots of false alarms in decision sub plot. Decision threshold is "*d*=0.72" in above analysis. We have applied the LS analysis on attack free data-1, Figure 5.4. is the results.

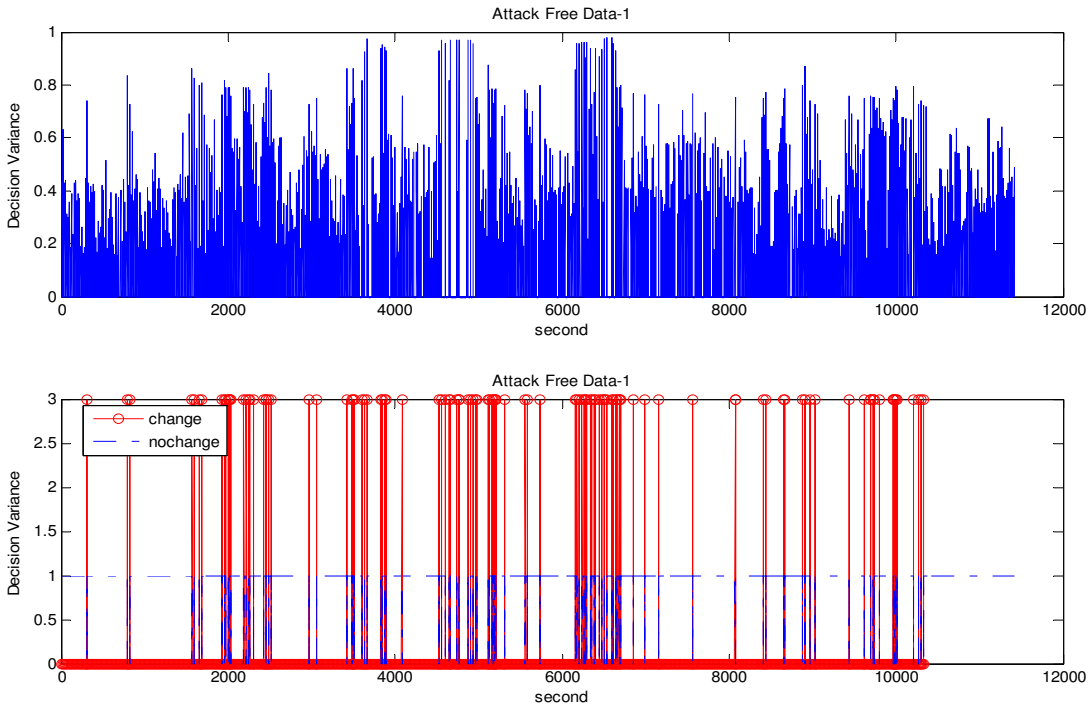

Figure 5.5. Attack free data-1

Figure 5.4. and 5.5. have showed us that there are lots of false alarms and there is error on LS analysis or our Test data, or our test data is not good for LS analysis. While we were running Matlab implementation there were lots of "Rank deficient" warnings in Matlab. This may be because of test data or LS structure. Because of these reasons this implementation did not work properly and it did not give meaningful results.

5.3. Yule Walker Method

To guarantee a valid output, we must set the Estimation order parameter to be less than or equal to half the input vector length the Yule-Walker AR.

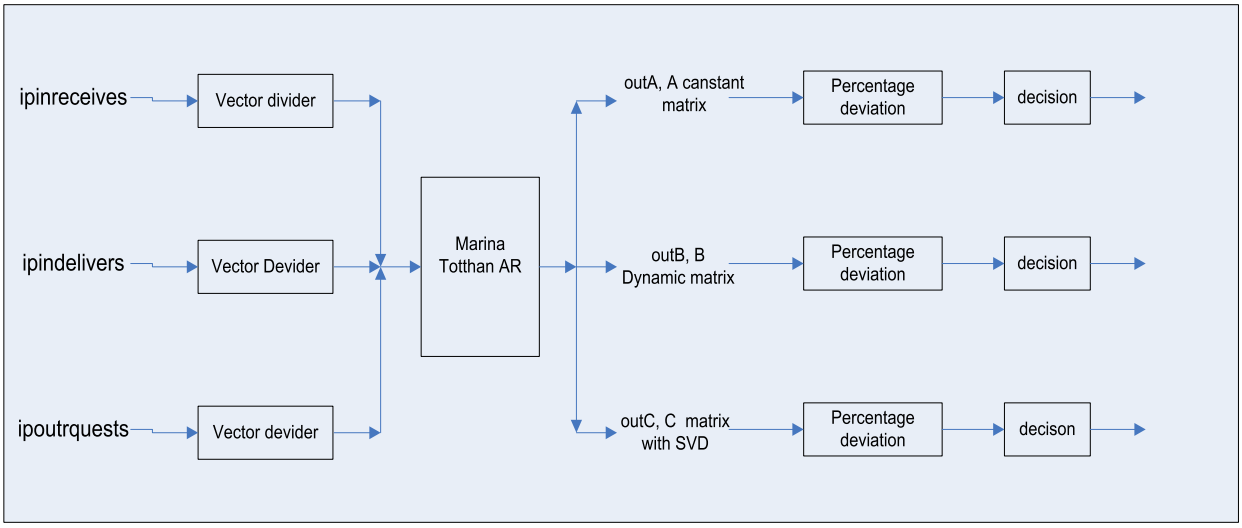

Figure 5.6. AR model block diagram

5.3.1. Results for Data with Attacks, AR(1)

We have performed same AR(1) analysis with Yule-Walker Estimator method, this estimator's matlab codes also written by R. Moses. Yule-Walker method gives us more accurate results than Least Squares as we can see from Figure 5.7. and 5.8. We can detect three attacks in the data, but beside this some false alarms could be seen. These false alarms are because of the sensitivity or deepness of AR analysis.

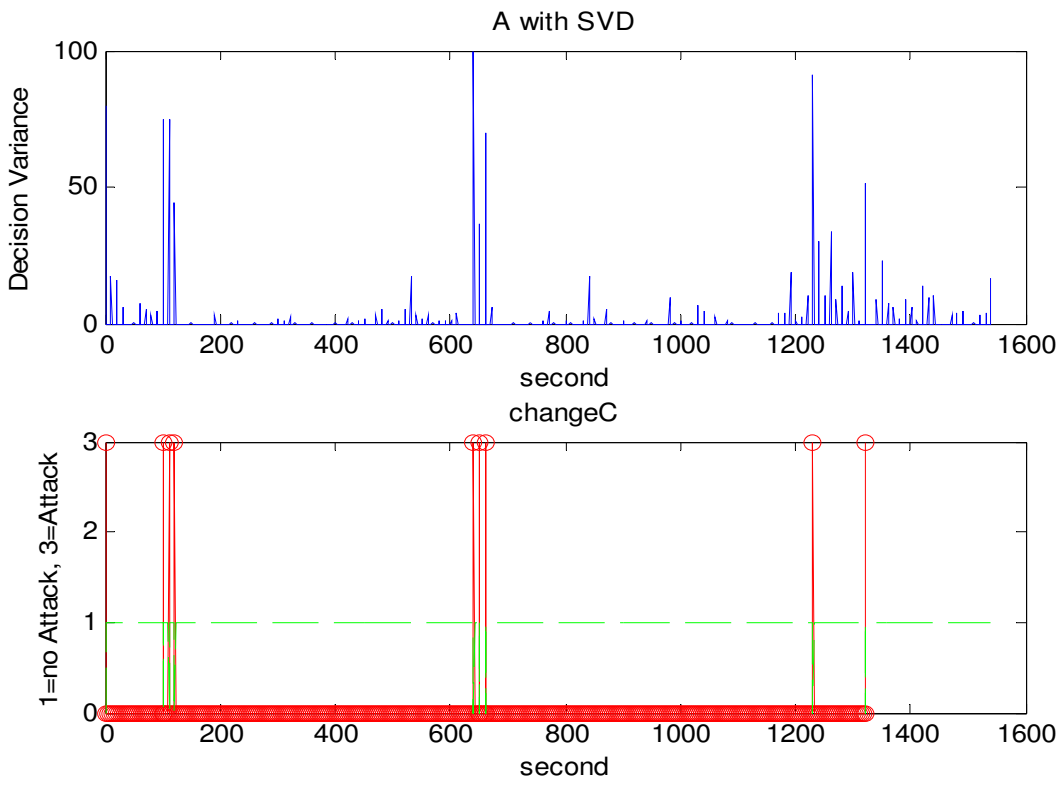

Figure 5.7. Data with Attacks, SVD for A operator matrix, th=36.1

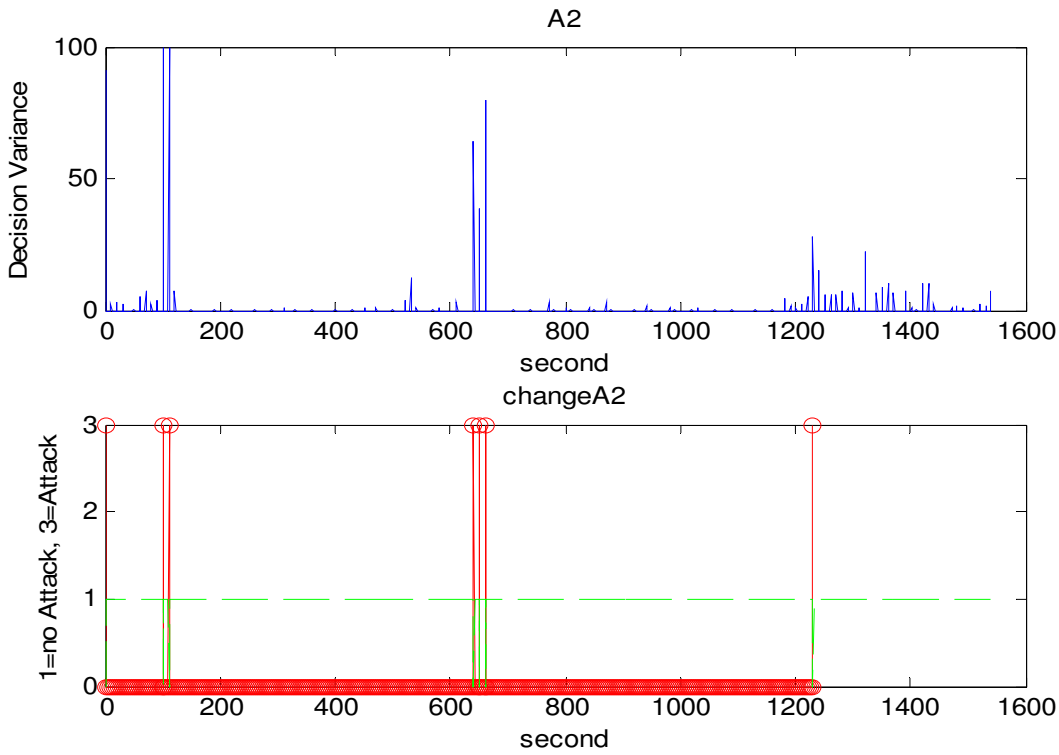

Figure 5.8. Data with attacks, A2 method for A operator matrix th=24.4

5.3.2. Results for Attack Free Data-1 AR(1)

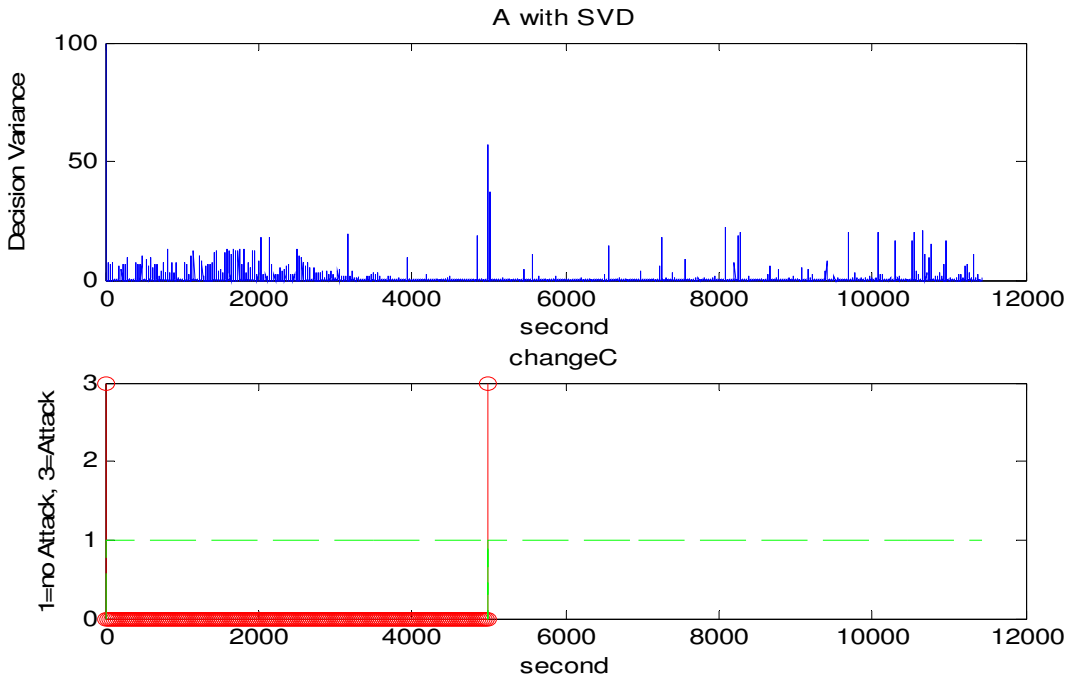

Figure 5.9. Attack free data-1, SVD for A operator matrix, th=44.9

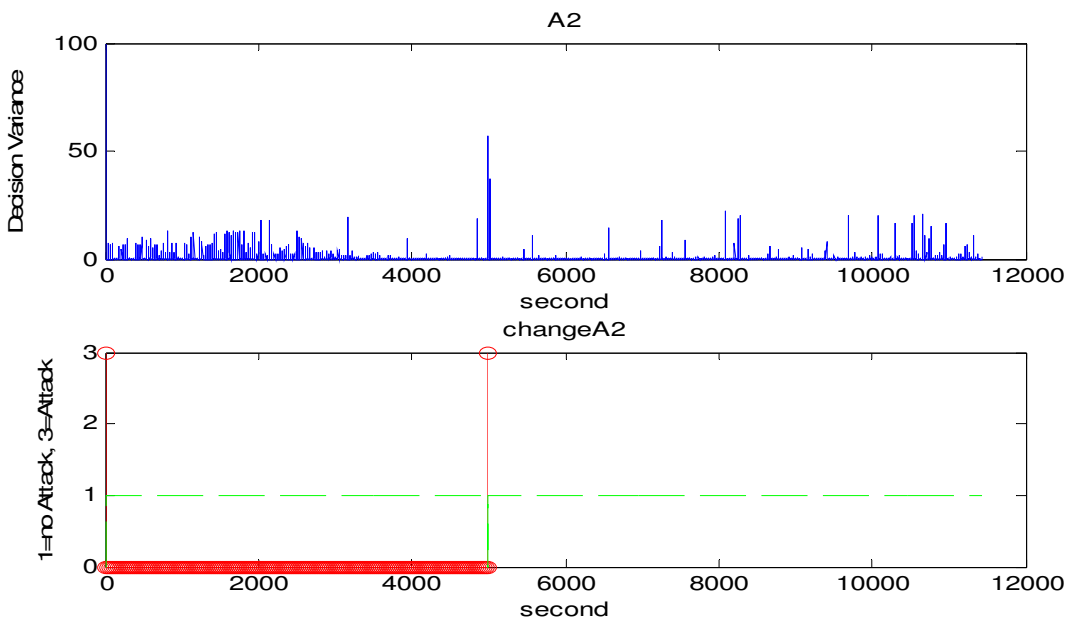

Figure 5.10. Attack free data-1, A2 method for A operator matrix th=44.9

All the Yule-Walker related Figures show that SVD and *A2* type *A* operator matrix have better detection rate than others. Because of that we only put these figures, we have also tested *A* static matrix and *A* with "corrcoef" types of *A* operator matrix. SVD and *A2*

type operator matrixes are flexible, and they have better detection performance. Figure 5.9. and 5.10. have showed us that Yule-Walker method is working properly and giving meaningful results. There is only one false alarm in the Figure 5.9. and 5.10., this also shows that the Yule-Walker method was working with attack free data-1 properly.

5.4. Modified Yule-Walker Method

5.4.1. Results for Data with Attacks AR(1)

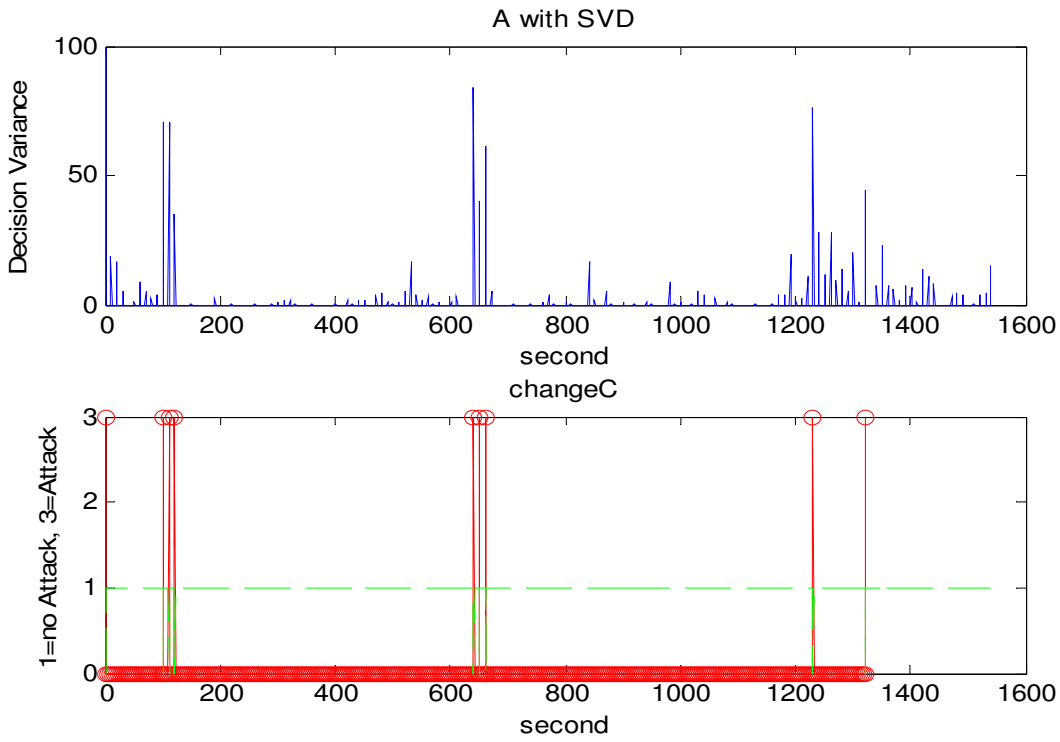

Figure 5.11. Data with attacks, SVD for A operator matrix, th=34

All the Modified Yule-Walker related Figures show that there is not a big performance differences between Modified Yule Walker and Yule Walker Method with our datasets. We have shown only Figure 5.11. in order to verify this expression. Because of this we did not show all other results.

"For small or medium samples lengths, Y-W and LS may behave differently, Y-W method is always guaranteed to be stable, whereas LS model may be unstable." Petre Stoica in [6].

5.5. Dataset-2

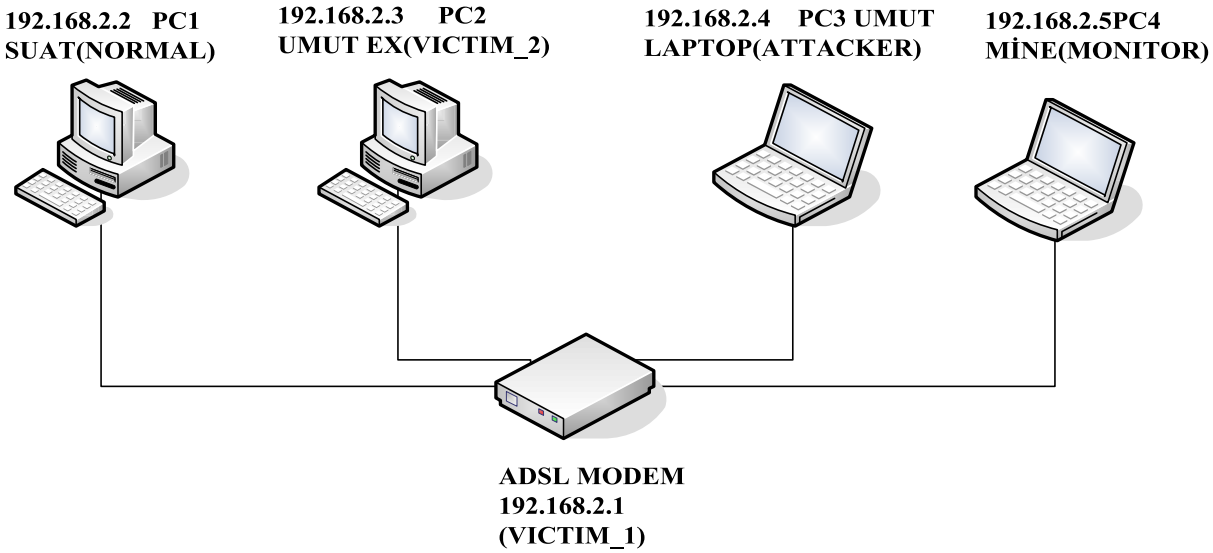

Figure 5.12. Dataset-2 network diagram

We have collected nine different data and their names are test2_02, test3_02, test1_03, test2_03, test4_03, test6_03, test7_03, test8_03, and test9_03. Each test is includes 42 minutes data and each test includes 500 steps of data. The data includes the variables IPinreceives, IPindelivers, IPoutrequests, and Ifinoctets. The important property of this dataset is ADSL modem was working as a switch. We use ADSL modem as a four port 100 MBits/sec switch and to connect to Internet. We have collected data under low traffic; these are test2_02, test3_02, test1_03, test2_03, and test4_03. we have collected data under high traffic also, these are; test6_03, test7_03, test8_03, and test9_03. We have used SolarWinds SNMP Real Time Graph program to collect these data. We have collected data with poll interval five seconds. We have used SolarWinds SNMP Brute Force Attack and Port Scanner to perform attacks. We have added all the attack details and detailed performance tables to Appendix.

5.6. Dataset-3

We have collected eight different data their names are test1_21, test2_21, test3_21, and test4_21 we have called these four Part1. Others are test5_21, test6_21, test7_21, and test8_21 we have called these four Part2. Each test is includes 42 minutes data and each test includes 500 steps of data. The data includes the variables IPinreceives, IPindelivers, IPoutrequests, and Ifinoctets. We have collected data with poll interval five seconds. The important property of this dataset is ADSL modem was working like a Router and Switch. ADSL modem was working as a Router between Blue-Net and Red-Net. ADSL modem was working as a Switch in Red-Net and same as in Blue-Net. By this separation we try to understand the behaviors of two different working modes. Test1_21, test2_21, test7_21 and test8_21 data were collected setup was working in Network-1 mode. Test3_21, test4_21, test5_21 and test6_21 data were collected setup was working in Network-2 mode. We have used SolarWinds SNMP Real Time Graph program to collect these data. We have used SolarWinds SNMP Brute Force Attack and Port Scanner to perform attacks. We have added all the details of attacks and data and detailed performance tables to Appendix.

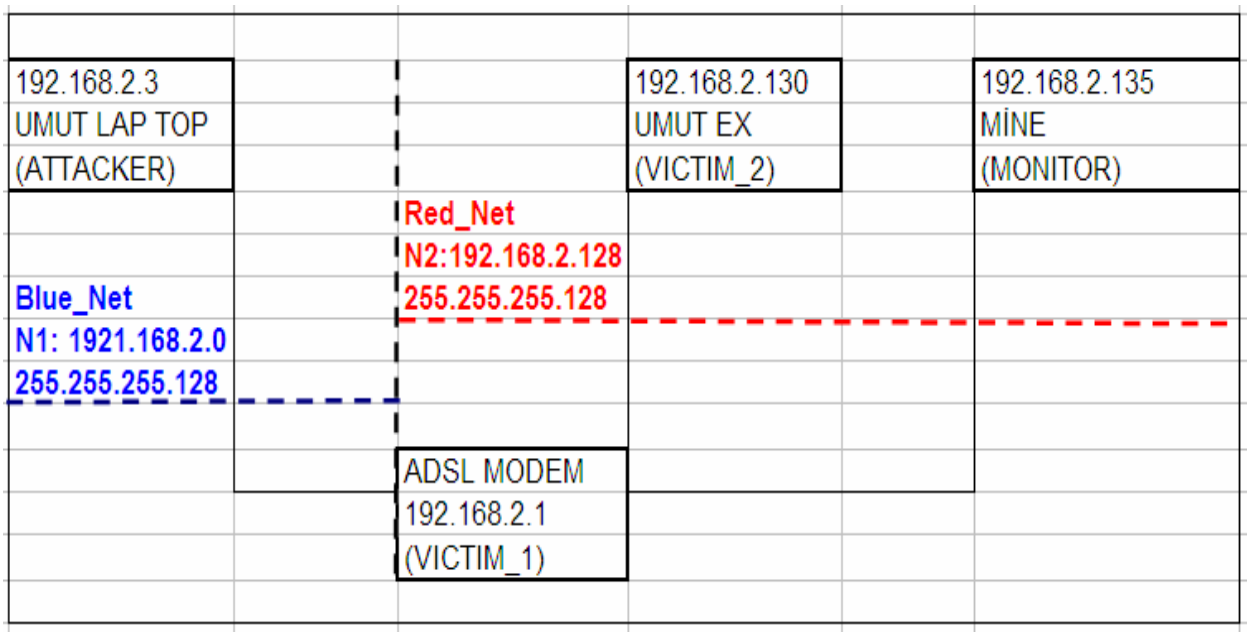

Figure 5.13. Dateset-3 network-1 diagram

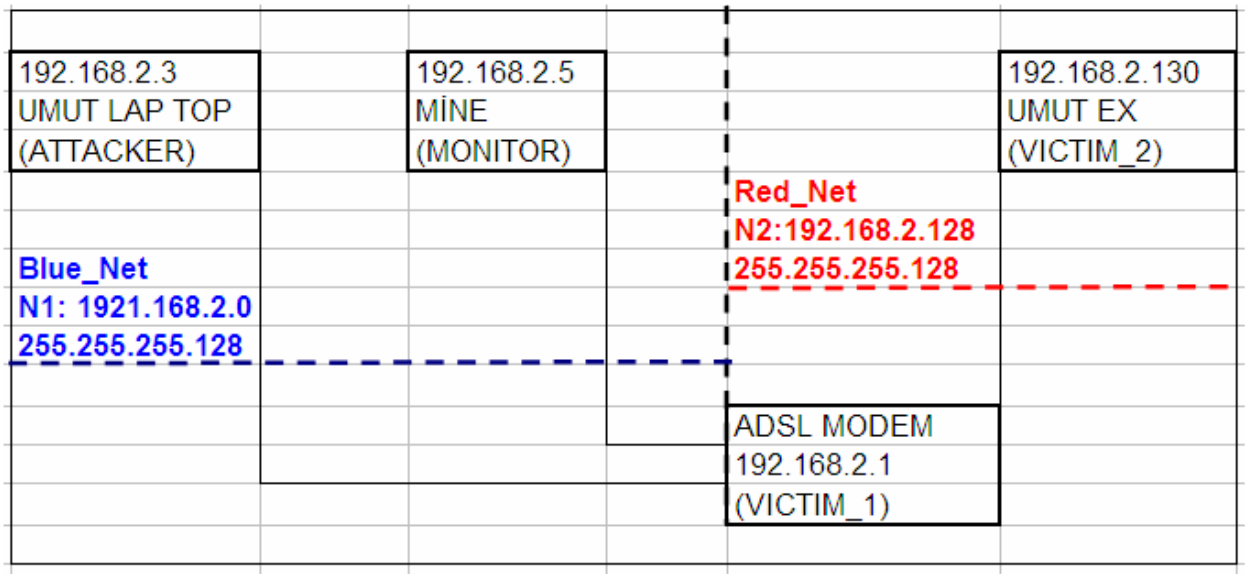

Figure 5.14. Dataset-3 network-2 diagram

5.7. Dataset-4

We have collected four different data their names are test1_29, test2_29, test3_29, and test4_29. We have also collected test1_20 and test2_20 under low traffic. Each test is includes 42 minutes data and each test includes 500 steps of data. The data includes the variables IPinreceives, IPindelivers, IPoutrequests, and Ifinoctets. We have collected data with poll interval five seconds. The important property of this dataset is ADSL modem was working like a Router. ADSL modem was working as a Router among Blue-Net, Red-Net, Orange-Net, and Green-Net. By this separation we try to understand the behaviors of Router working mode. We have used SolarWinds SNMP Real Time Graph program to collect these data. We have used SolarWinds SNMP Brute Force Attack and Port Scanner to perform attacks. We have added all the details of attacks and data and detailed performance tables to Appendix.

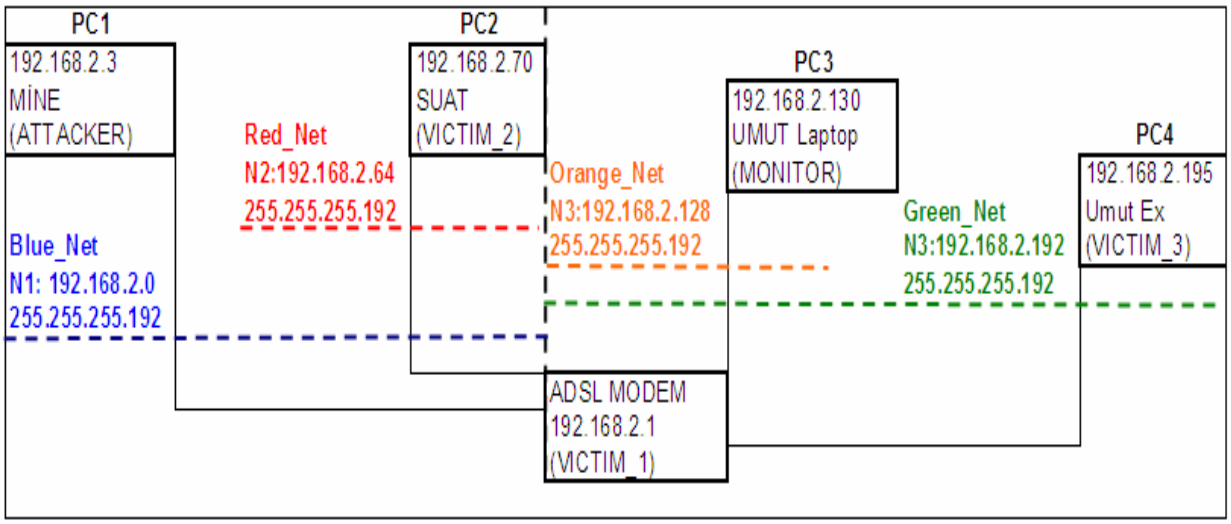

Figure 5.15. Dataset-4 network diagram

5.8. The Key Variables and Functions are Affecting the Decision Vector and Also Result

We will explain these variables and functions on "data with attacks" data set. We want show the key point of the analysis methods which are really positively affecting the detection performance.

5.8.1. Likelihood Ratio η Effects on Decision

We have already defined the Likelihood Ratio η in methodology section. Now we will show the effects of η to detection performance. Let us remind η once more.

$$\eta = \frac{\hat{\sigma}_R^{-N'_R}\hat{\sigma}_S^{-N'_S}}{\hat{\sigma}_R^{-N'_R}\hat{\sigma}_S^{-N'_S} + \hat{\sigma}_P^{-(N'_R+N'_S)}} \tag{5.1}$$

AR analysis works on ten step size which means we take ten samples from our time series data and calculate variance of this window. We use ten samples for test and learning window. We use twenty samples for pooled window and we shift the window ten by ten that means there is no overlap in our model. We will show test variance, learning variance, and pooled variance of our time series data which are ten times smaller than input vector because of the above reason.

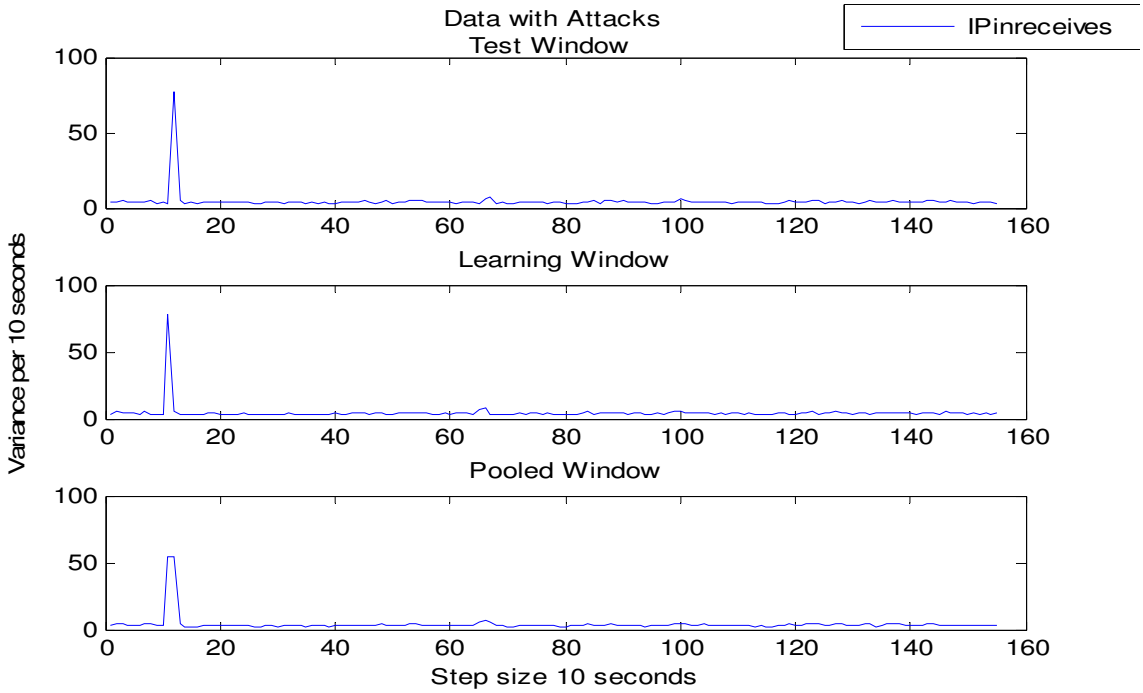

Figure 5.16. Variances of IPinreceives

In Figure 5.16. we could not catch the second attack. Variance plots of IPindelivers and IPoutrequest are same fashion in other words we could not catch the second attack in other two variables either. Because of this we will not show all the variance plots. We will show all three Likelihood Ratio η in Figure 5.17. In Figure 5.17. we could easily see the second attack, but beside second attack there are other spikes which may yield false alarms. From the Figure 5.17. we understand that Likelihood Ratio is affecting the decision performance in positive way and it extracts the hidden information in input vectors which yields us a sensitive and deeper analysis method, but in some cases it is too deep analysis. We try to adjust these sensitivity and deepness of AR analysis via Wavelet Model in next sections.

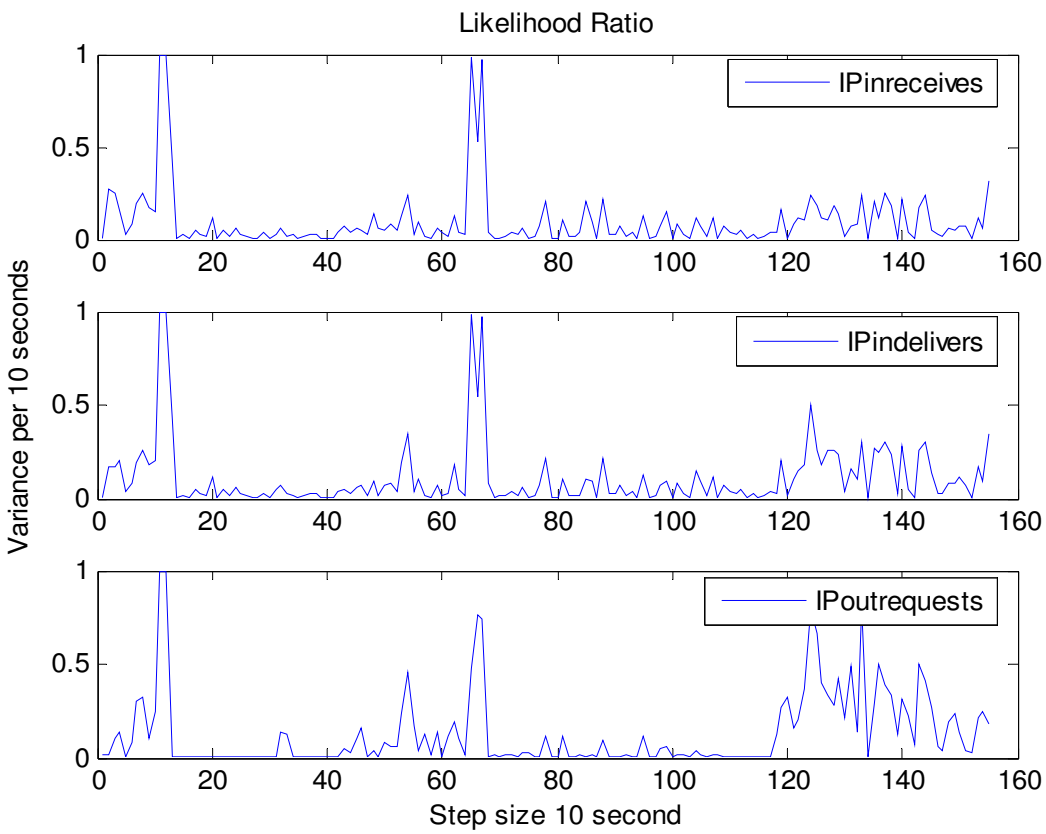

Figure 5.17. Likelihood ratios of variables

5.8.2. *A* Operator Matrix Effects on Decision

We have three variables, may be we can more with Wavelet Analysis. We can not give a good decision with multiple variables on decision step for one issue. So we need to combine them in to one variable. *Totthan et al* have suggested an *A* operator matrix to combine the multiple decision variables. We have also proposed new models for creating *A* operator matrix. We will try to show the benefits of *A* operator matrix. We have used four different type *A* operator matrix which are, *A* Static 3x3 Matrix, *A* with Corrcoef, *A* with SVD, and *A2*. *A* static Matrix has suggested in [4], and *A2* type operator matrix has suggested in [1]. Figure 5.18. shows us all four type *A* operator matrix results on Data with Attacks dataset. *A* operator matrix is combining the multiple input to one decision variable.

Beside this it reduces the noise effect. We can see noise in Likelihood ratio plots easily and it has a significant effect on decision variable. But, after *A* operator matrix combines the variables it also reduces the noise effect on decision variable this increase the detector performance in positive way.

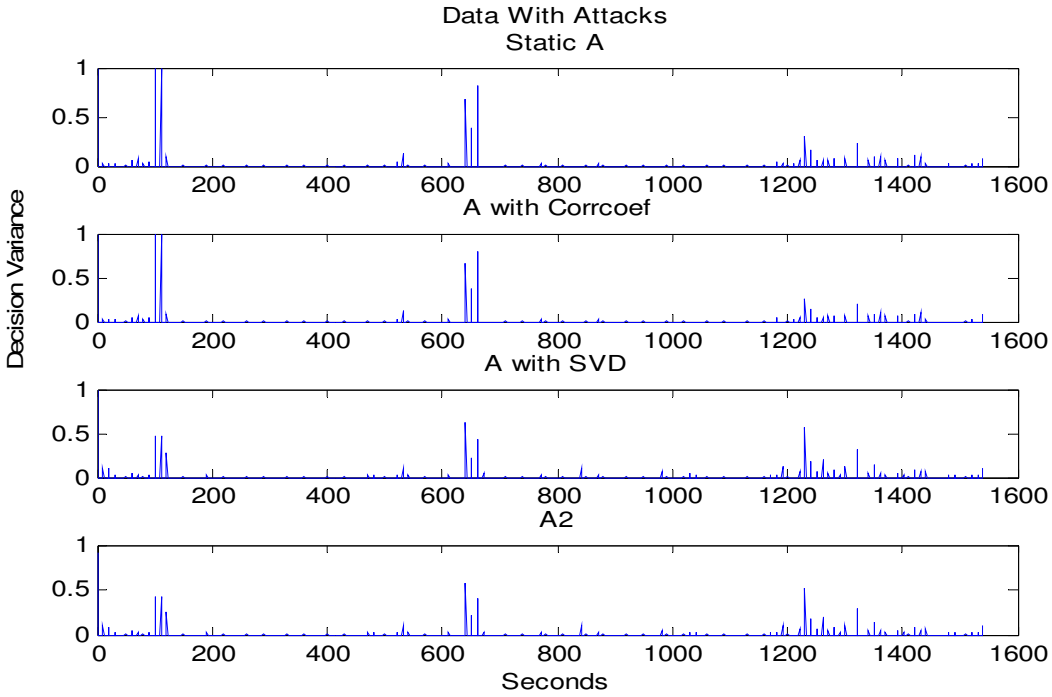

Figure 5.18. Four type A operator matrix results

6. RESULTS-WAVELET MODEL

6.1. Wavelet-Modulus Maxima Model

6.1.1. Data with Attacks Dataset

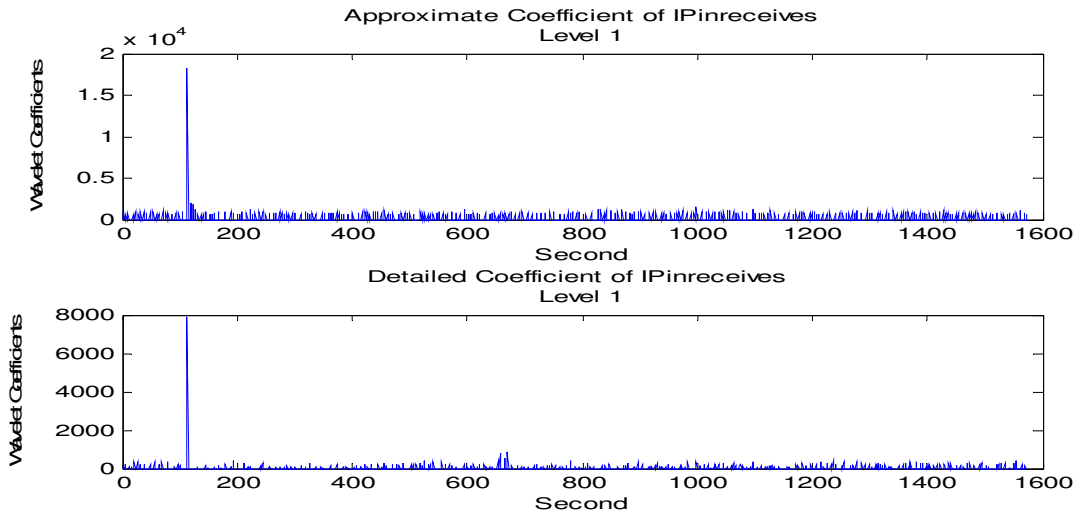

Figure 6.1. Data with attacks dataset/IPinreceives wavelet level-1 coefficients

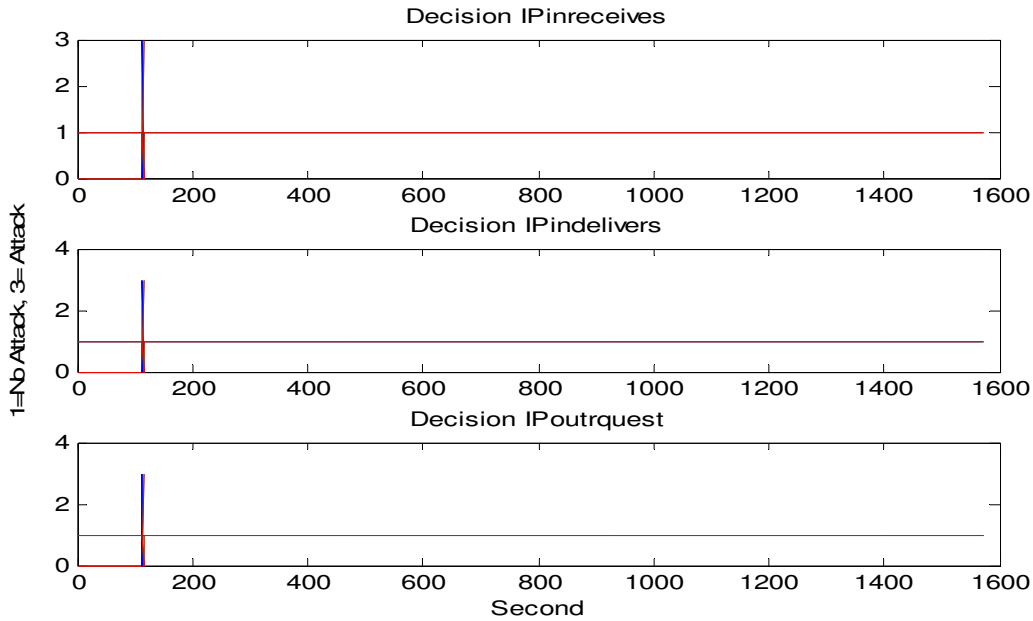

Figure 6.2. Decision on data with attacks dataset approximate coefficients

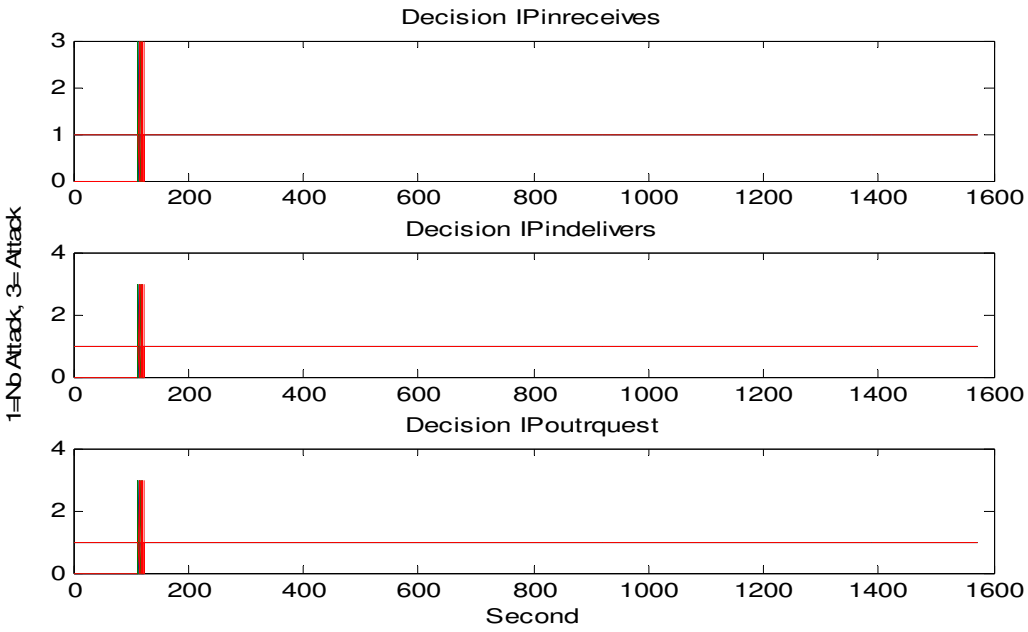

Figure 6.3. Decision on data with attacks dataset detailed coefficients

We have used Level three Wavelet analysis and Coiflets order five Mother Wavelet for Wavelet analysis. We have test some other Mother Wavelets like, Haar, Daubechies orders 2-3-4-5-6-7-8, Symlets orders 2-3-4-5-6, Coiflets orders 1-2-3-4-5, and DMeyer, BiorSplines 1.1. We observe that Coiflets order five Mother Wavelet was the best for our dataset's Wavelet Analysis. After we have applied the wavelet analysis we have fed the wavelet coefficients to "modulus-maxima" function. Modulus-Maxima function can find the sharp variation points in data [13]. But, we could not catch the hidden attacks in our datasets as seen in Figure 6.1., 6.2., and 6.3. We can see only one attack in the decision section. Because of this we have Miss-Alarms, and this Modulus Maxima detector has not enough sensitivity to detect hidden attacks in datasets. If an attack has a significant change on traffic values we could detect it easily. We will not show all the approximate and detailed coefficients of other datasets we will only show the decision plots of them.

We have seen two false alarms on Decision IPinreceives on Detailed Coefficients. Since detailed coefficients includes some spikes, these spikes causes false alarm in decision section. Because of this reason we are using approximate coefficients of wavelet analysis.

6.2. Wavelet-AR Model.

We have performed the Wavelet-AR analysis in two ways.

First one is, we have applied AR model on each levels approximate coefficients, it yields more than one decision variable if we use higher than level one. Let's say we are using level 3 for wavelet analysis which yields three decision variables.

Second way is, we are combining all approximate coefficients of all levels and create a big *A* operator matrix.

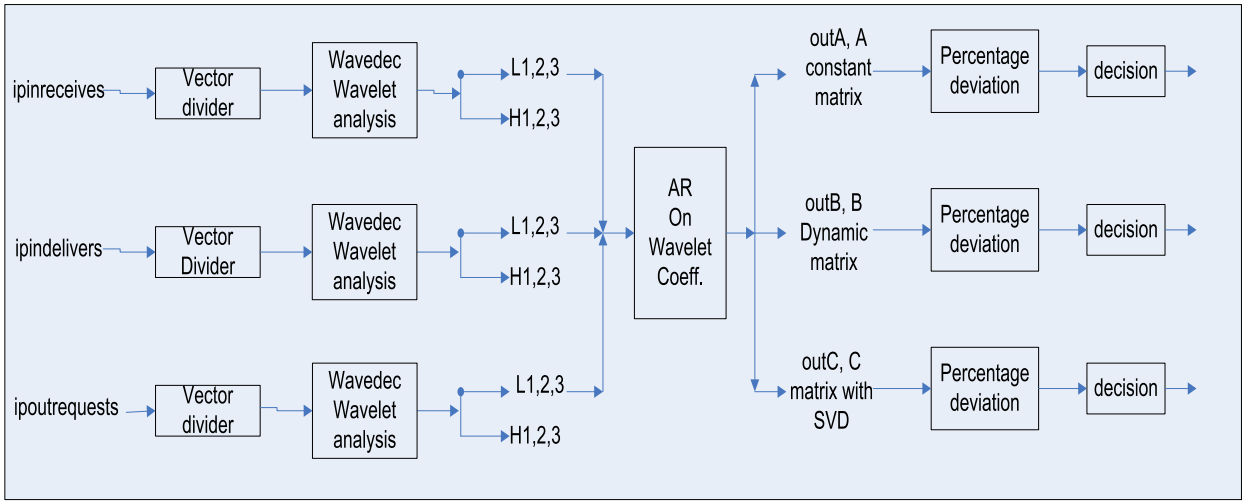

Figure 6.4. 3 input Wavelet-AR model

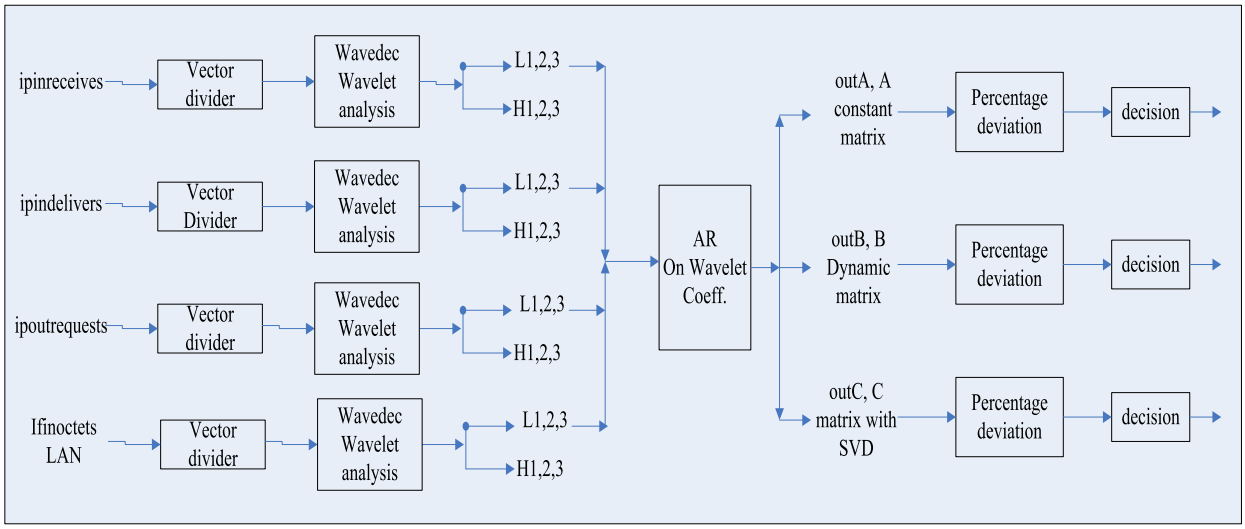

Figure 6.5. 4 input Wavelet-AR model

6.2.1. Wavelet-AR Analysis Level by Level

In Figure 6.6. and 6.7. we have used different colors to represent the levels. Attack colors are; red is representing the level-1, yellow is representing the level-2 and blue is representing the level-3 and green is representing the normal condition on the third figures in all Figure 6.6., 6.7., and 6.8.. As we can see from the Figure 6.6. and 6.7. first attack is significant in all levels, but second and third attack are significant in only level-1 because of this we do not need to use level-2 and level-3, and also this level reduction positively affect the computational burden of the Wavelet-AR model.

Data with attack dataset and attack free dataset are shown.

- Data with Attacks Dataset AR(1)

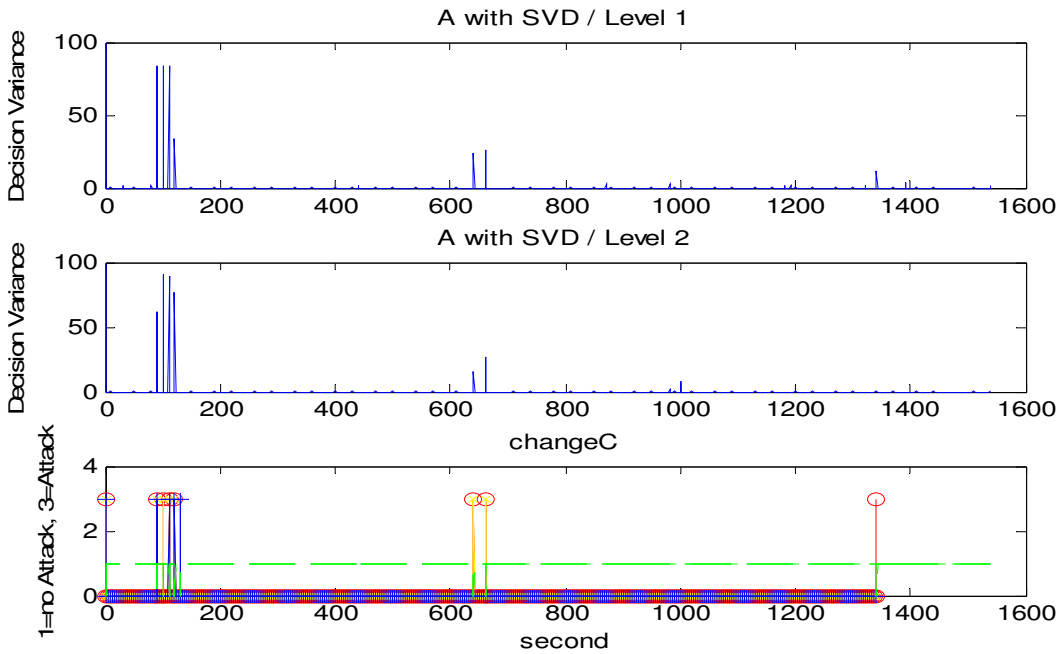

Figure 6.6. Data with attacks dataset, A with SVD 3x3, level 1, 2, 3 analyzed level by level th:9.3

"If the level is large then coarse approximation of signal is achieved, so details are neglected. If the level is low a detailed approximation of signal is achieved." H.J. Barnard in [10].

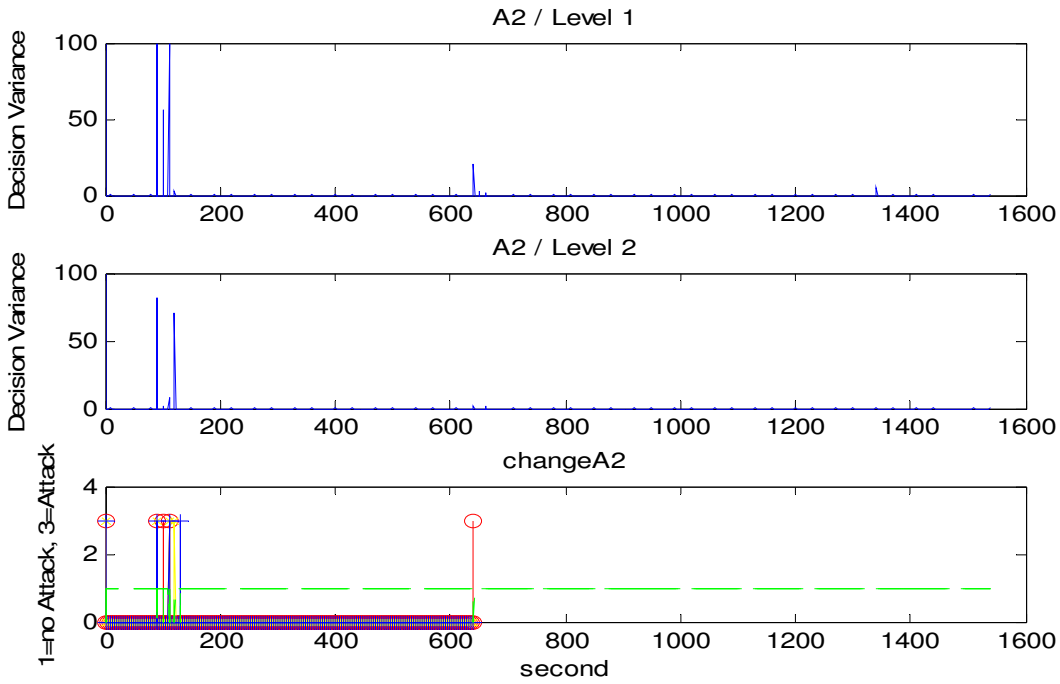

Figure 6.7. Data with attacks dataset, A2 type matrix 3x3 , level 1, 2, 3 analyzed level by level th:8.9

As we can see from the Figure 6.6. and 6.7. SVD method has better detection rate than other. With SVD method we can detect the second and third attacks, but in other methods we can not detect the third attack. Although the threshold in SVD method is greater than *A2*'s threshold, SVD shows better detection rate.

- Attack Free Data-1 Dataset AR(1)

While we were plotting the levels of Wavelet, we use Red for Level-1, Yellow for Level-2, Blue for Level-3 and Green for normal traffic. As we can see from Figures 6.8. there are some spikes in Decision Variance plots this yields some false alarms. With attack free data set we want to show the false alarm rate of our system. Because high per cent of network operation time network does not contains attacks, an IDS system should have fewer false alarms; moreover false alarm rate of a system should be in acceptable limits.

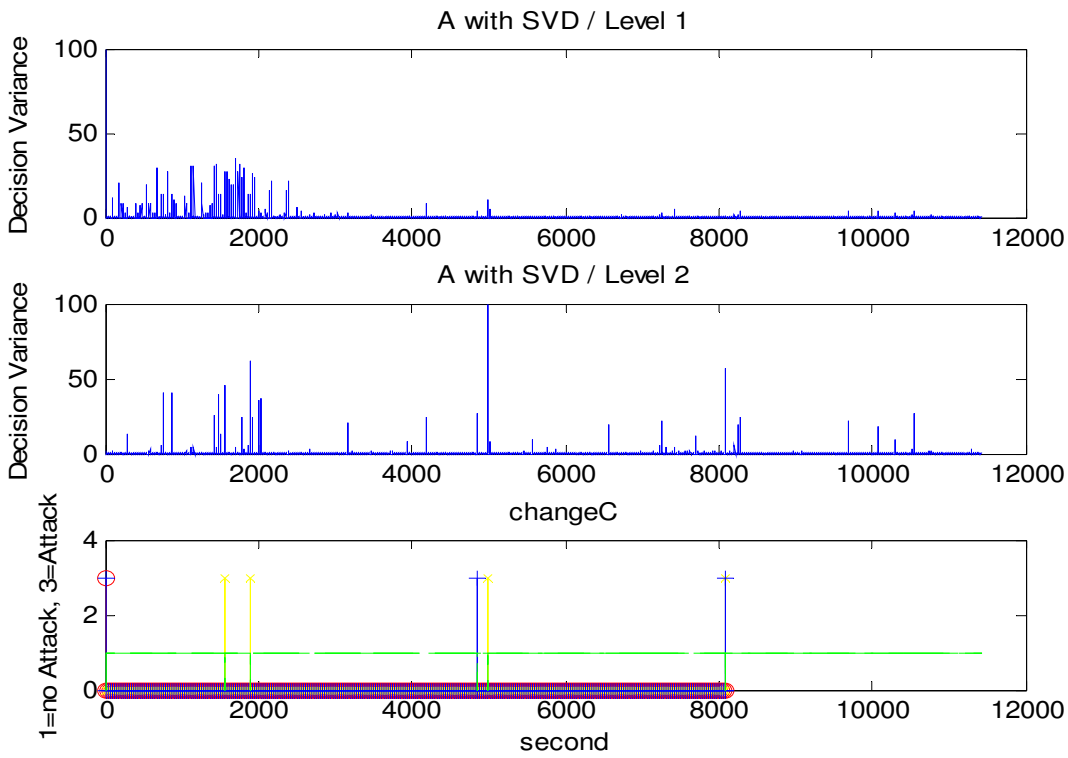

Figure 6.8. Attack free data-1 dataset, A with SVD, level 1, 2, 3 analyzed level by level, th:45

6.3. Wavelet-AR All Levels Together

In this section we will show the Wavelet-AR analysis on all levels together. This yields *9x9, 6x6*, and *3x3 A* operator matrix. If we perform this model on level-1-2-3 and level-1-2 it reduce the detection rate, but we do not want this reduction we want to reduce the false alarms without reducing the detection rate so best level is level-1 only.

6.3.1. Data with Attacks Dataset AR(1) Level 1, 2, 3

As we can see from Figure 6.10. operator matrix A(SVD) and B(corrcoef) they are totally different. This difference comes from their definition. From the results SVD type *A* operator matrix has better detection performance than others in two cases level by level and all levels together. Figure 6.9. shows that all levels together type analysis reducing the detection rate.

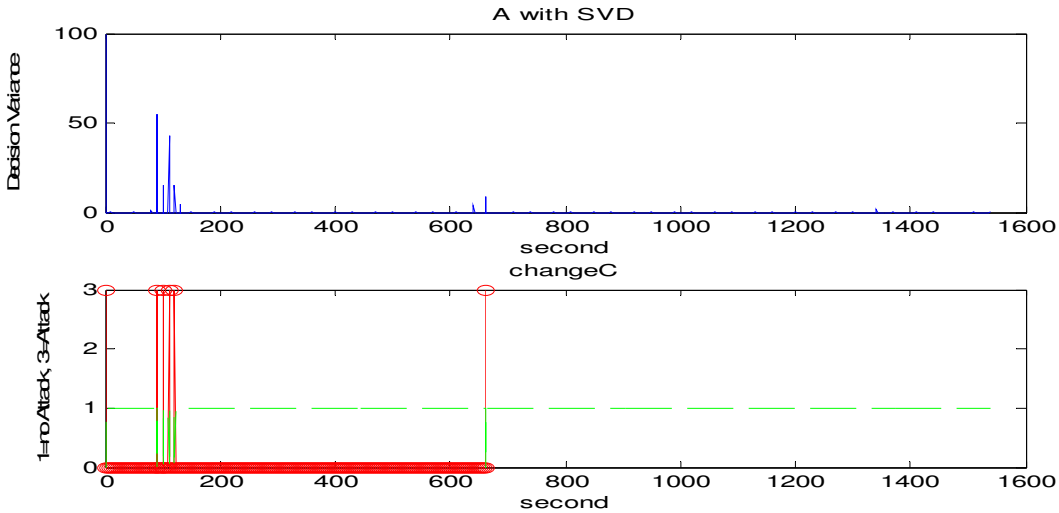

Figure 6.9. Data with attacks dataset, A with SVD 9x9, level 1, 2, 3 analyzed together th:5.02

We have added two sample *A* operator matrixes, first one is *A* operator matrix with SVD method and *9x9*, second one is "corrcoef" method and it is also *9x9* in Figure 6.10.

$$A=\begin{bmatrix} 2.6363 & -0.0614 & 3.3705 & -1.2170 & 0.3437 & -1.1163 & -1.8360 & -0.5863 & 0.4665 \\ 0.5977 & -0.1617 & 0.2912 & 0.0086 & 0.0567 & 0.1485 & -0.3680 & -0.2539 & -0.3160 \\ -0.9571 & -2.9353 & 0.8590 & -0.4444 & -0.0372 & 1.8771 & -0.1222 & 3.7745 & -1.2062 \\ -0.4334 & -0.2961 & -0.9760 & 3.9508 & -1.5878 & 0.4071 & -2.0588 & 0.9453 & 0.1210 \\ 0.2750 & 0.2102 & -0.2074 & 0.6558 & -0.0418 & -0.3783 & -0.2777 & -0.8405 & -0.0277 \\ -0.9410 & -2.0154 & 0.6153 & 0.0896 & 0.2235 & 1.5475 & -0.1125 & 2.8472 & -0.8863 \\ -1.6883 & 0.0219 & -1.5175 & -2.0575 & 0.8559 & 0.6540 & 3.1226 & 0.5911 & -0.8306 \\ -1.1951 & 0.5214 & -0.6482 & -1.0857 & 0.0627 & 0.1702 & 1.8194 & -0.1653 & 0.3882 \\ 1.9642 & 4.2971 & -1.4758 & 0.4443 & 0.0506 & -3.0372 & 0.4109 & -5.9424 & 2.2191 \end{bmatrix}$$

$$B=\begin{bmatrix} 0.1111 & 0.0888 & 0.0715 & 0.1111 & 0.0887 & 0.0714 & 0.1111 & 0.0888 & 0.0715 \\ 0.0888 & 0.1111 & 0.0894 & 0.0887 & 0.1111 & 0.0894 & 0.0888 & 0.1111 & 0.0894 \\ 0.0715 & 0.0894 & 0.1111 & 0.0714 & 0.0894 & 0.1111 & 0.0715 & 0.0894 & 0.1111 \\ 0.1111 & 0.0887 & 0.0714 & 0.1111 & 0.0887 & 0.0714 & 0.1111 & 0.0887 & 0.0714 \\ 0.0887 & 0.1111 & 0.0894 & 0.0887 & 0.1111 & 0.0894 & 0.0887 & 0.1111 & 0.0894 \\ 0.0714 & 0.0894 & 0.1111 & 0.0714 & 0.0894 & 0.1111 & 0.0714 & 0.0894 & 0.1111 \\ 0.1111 & 0.0888 & 0.0715 & 0.1111 & 0.0887 & 0.0714 & 0.1111 & 0.0888 & 0.0715 \\ 0.0888 & 0.1111 & 0.0894 & 0.0887 & 0.1111 & 0.0894 & 0.0888 & 0.1111 & 0.0894 \\ 0.0715 & 0.0894 & 0.1111 & 0.0714 & 0.0894 & 0.1111 & 0.0715 & 0.0894 & 0.1111 \end{bmatrix}$$

Figure 6.10. A:SVD and B:Corrcoef type A operator matrixes

6.3.2. Data with Attacks Dataset AR(1) Level 1.

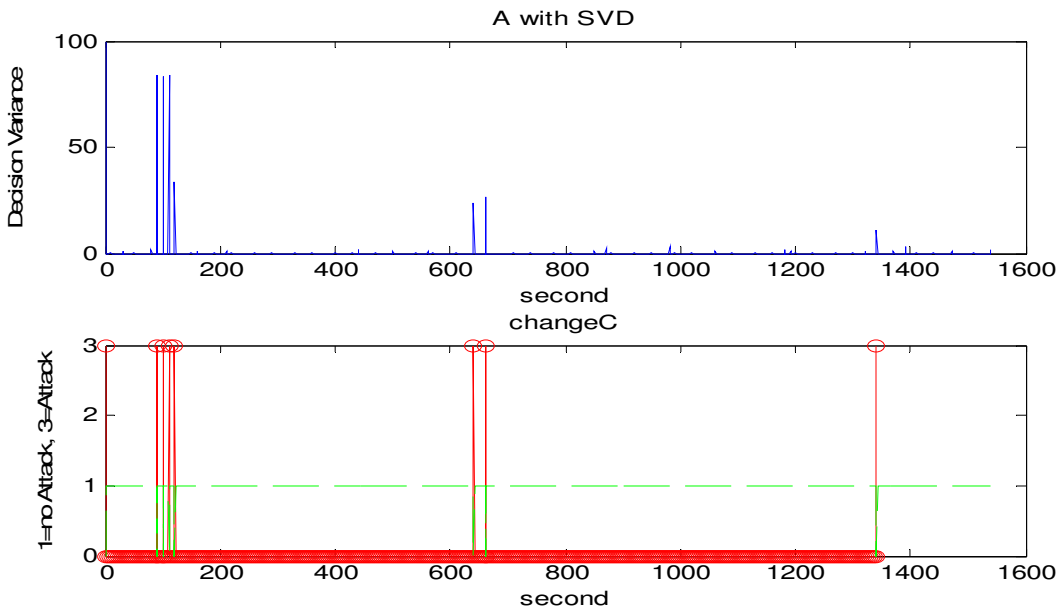

Figure 6.11. Data with attacks dataset, AR(1) A with SVD , level 1 analyzed th:8.4

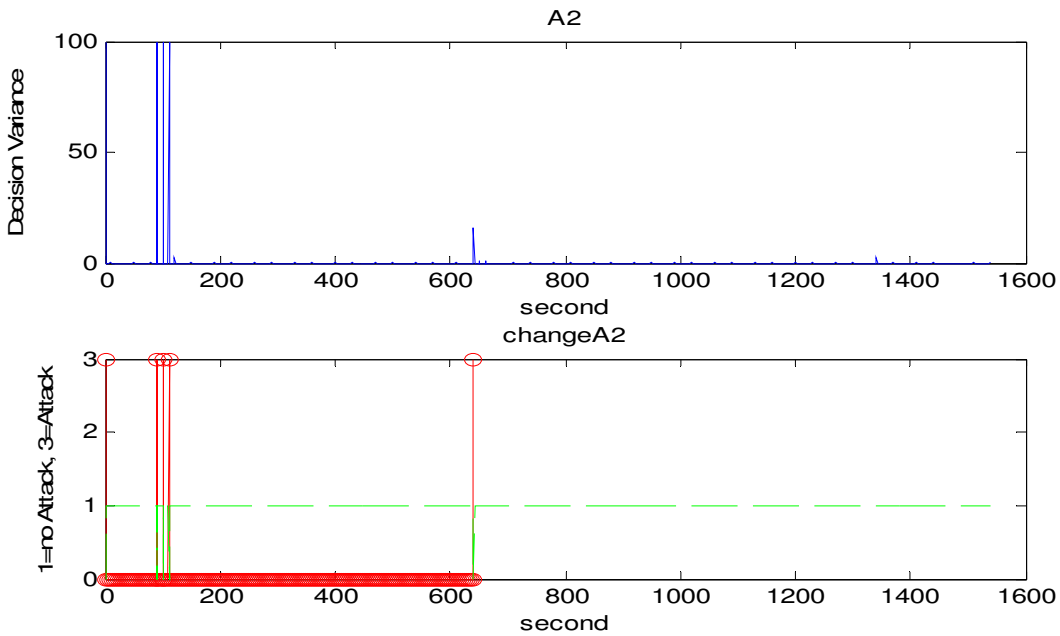

Figure 6.12. Data with attacks dataset, AR(1), A2 type, level 1 analyzed th:7.4

Figure 6.11. and 6.12. show that result of A with SVD, and *A2* type *A* operator matrixes, as we can see from figures, Figure 6.11. has better performance than Figure 6.12.. This also shows that SVD is a good way of creating *A* operator matrix. We have

added some examples of *A* operator matrix which are AR(1) means *3x3* and they have taken from Matlab operations. SVD type *A* operator matrix is really different from others.

Examples of *A* operator matrix with AR order one.

- $B(\text{Corcoeff}) = \begin{bmatrix} 0.3333 & 0.3332 & 0.3333 \\ 0.3332 & 0.3333 & 0.3332 \\ 0.3333 & 0.3332 & 0.3333 \end{bmatrix}$

- $A2\ (\text{A2 type}) = \begin{bmatrix} 0.3339 & 0.3353 & 0.3308 \\ 0.3353 & 0.3375 & 0.3329 \\ 0.3308 & 0.3329 & 0.3494 \end{bmatrix}$

- $A\ (\text{SVD}) = \begin{bmatrix} -12.7543 & 11.6596 & 3.0947 \\ 7.0819 & -15.8505 & 3.3410 \\ 5.1389 & 3.9169 & -6.8843 \end{bmatrix}$

6.3.3. Data with Attacks Dataset AR(3) Level 1

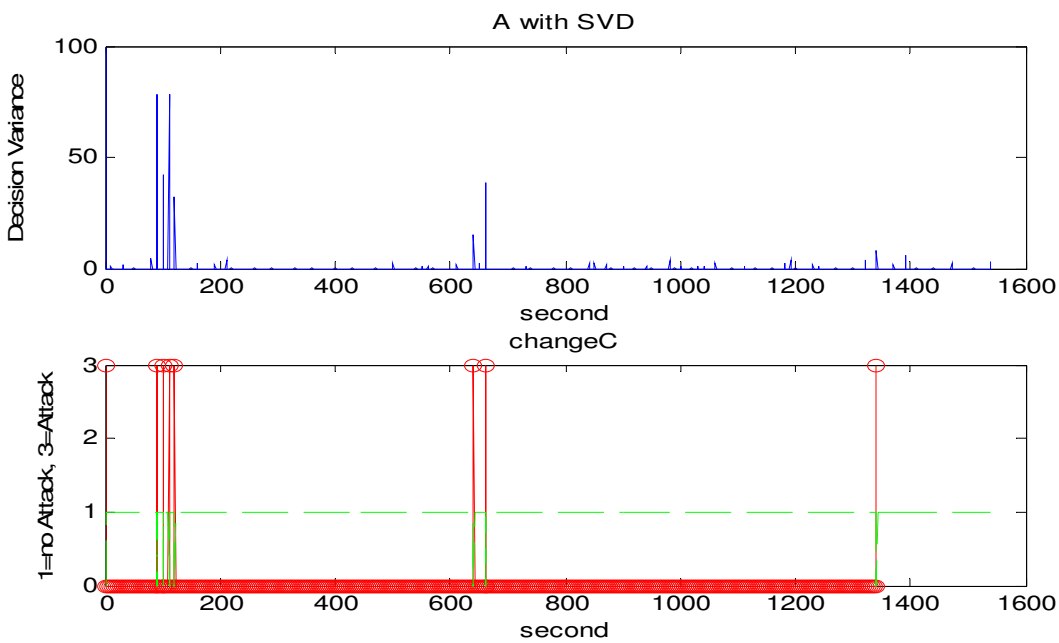

Figure 6.13. Data with attacks dataset, AR(3), A with SVD , level 1 analyzed th:7.7

Figures 6.13. shows us AR order three which adds some little improvement on Decision Variance can yield improvement on detection rate. As we can see from Figures 6.13. SVD type *A* operator matrix has good performance. AR order one is satisfying our needs so we do not need to use order three and order one has lower computation. We have added some examples of *A* operator matrix which are taken from Matlab operations. SVD type *A* operator matrix is really different from others and also from AR order one.

Examples of *A* operator matrix with AR order three

- $B(\text{Corrcoef}) = \begin{bmatrix} 0.3333 & 0.3332 & 0.3333 \\ 0.3332 & 0.3333 & 0.3332 \\ 0.3333 & 0.3332 & 0.3333 \end{bmatrix}$

- $A2(A2\ Type) = \begin{bmatrix} 0.3394 & 0.3398 & 0.3207 \\ 0.3398 & 0.3413 & 0.3222 \\ 0.3207 & 0.3222 & 0.3393 \end{bmatrix}$

- $A\ (\text{SVD}) = \begin{bmatrix} -10.0580 & 9.9392 & 2.1188 \\ 5.9598 & -12.9557 & 2.3528 \\ 3.7286 & 2.8137 & -5.0769 \end{bmatrix}$

6.3.4. Attack Free Data-1 Dataset AR(1) Level 1

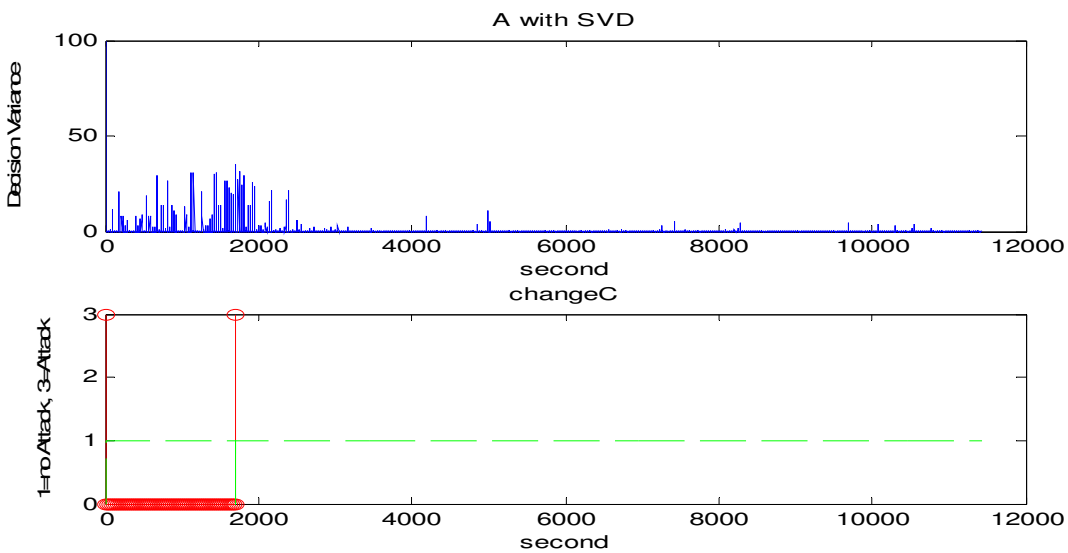

Figure 6.14. Attack free data-1 dataset AR(1), A with SVD , level-1 analyzed th:32.7

When we compare the results between AR(1) model and Wavelet level-1 AR(1) model we have fewer false alarms with Wavelet level-1 AR(1) model. When we compare the thresholds between two models AR model has greater threshold than Wavelet-AR model. For example *th(AR)= 38.4* and *th(Wavelet-AR)= 32.7*. Figure 6.14. has showed us Wavelet-AR model was working properly with attack free data.

6.4. Matlab Codes for Wavelet Analysis

We have used the following matlab codes for AR and Wavelet-AR analysis; we add Matlab codes to Appendix.

- **wavelet_modmax_perdev1.m**
    1. vectordivider.m
    2. perdeviation1.m
    3. decision.m
    4. modulusmaxima.m

- **waveletdb6_mTotthan.m (Level by Level Analysis)**
    1. eyule_wavelet_ext.m (Four type A calculation)
    2. yule_ar.m
    3. yulewalker.m (Yule-Walker method for AR spectral estimation, by R. Moses)
    4. vectordivider.m
    5. perdeviation.m
    6. perdeviation1.m
    7. decision1.m

- **wavelet_mTotthan_dec_ver3.m (All Leves Together Analysis)**
    1. eyule_wavelet_ver3.m (AR model)
    2. eyule_wavelet_ext2.m (Four type A calculation)
    3. yule_ar.m
    4. yulewalker.m (Yule-Walker method for AR spectral estimation, by R. Moses)

5. yule_ar_ver1.m
6. vectordivider.m
7. perdeviation1.m
8. decision2.m

6.5. The Key Variables and Functions are Affecting the Decision Vector and Also Results

We will explain this variables and functions on “data with attacks” dataset.

6.5.1. Likelihood Ratio η Effects on Decision

We have already defined the Likelihood Ratio η in methodology section. Now we will show the effects of η. Let remind η once more.

$$\eta = \frac{\hat{\sigma}_R^{-N'_R}\hat{\sigma}_S^{-N'_S}}{\hat{\sigma}_R^{-N'_R}\hat{\sigma}_S^{-N'_S} + \hat{\sigma}_P^{-(N'_R+N'_S)}} \tag{6.1}$$

AR analysis work on ten step size which means we take ten samples from our time series data and estimate variance of this window. We use ten samples for test and learning window. We use twenty samples for pooled window and we shift the window ten by ten there is no overlap in our model. In Figure 6.15. we could see test variance, learning variance, and pooled variance time series which are ten times smaller than input vector because of the above reason. We have also showed Variance of Approximate Coefficients of IPinreceives in Figure 6.16. From Figure 6.15. and 6.16. we could not detect the second and third attacks. This means that only variance changes were not enough for detection of hidden attacks.

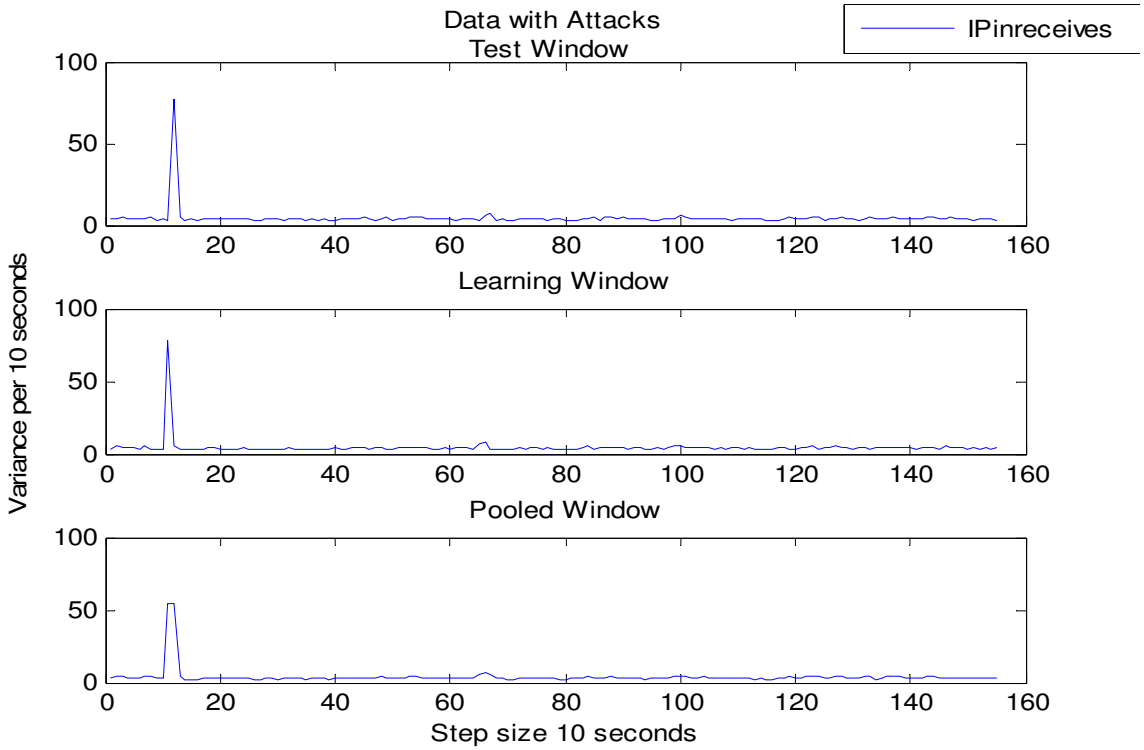

Figure 6.15. Variances of IPinreceives

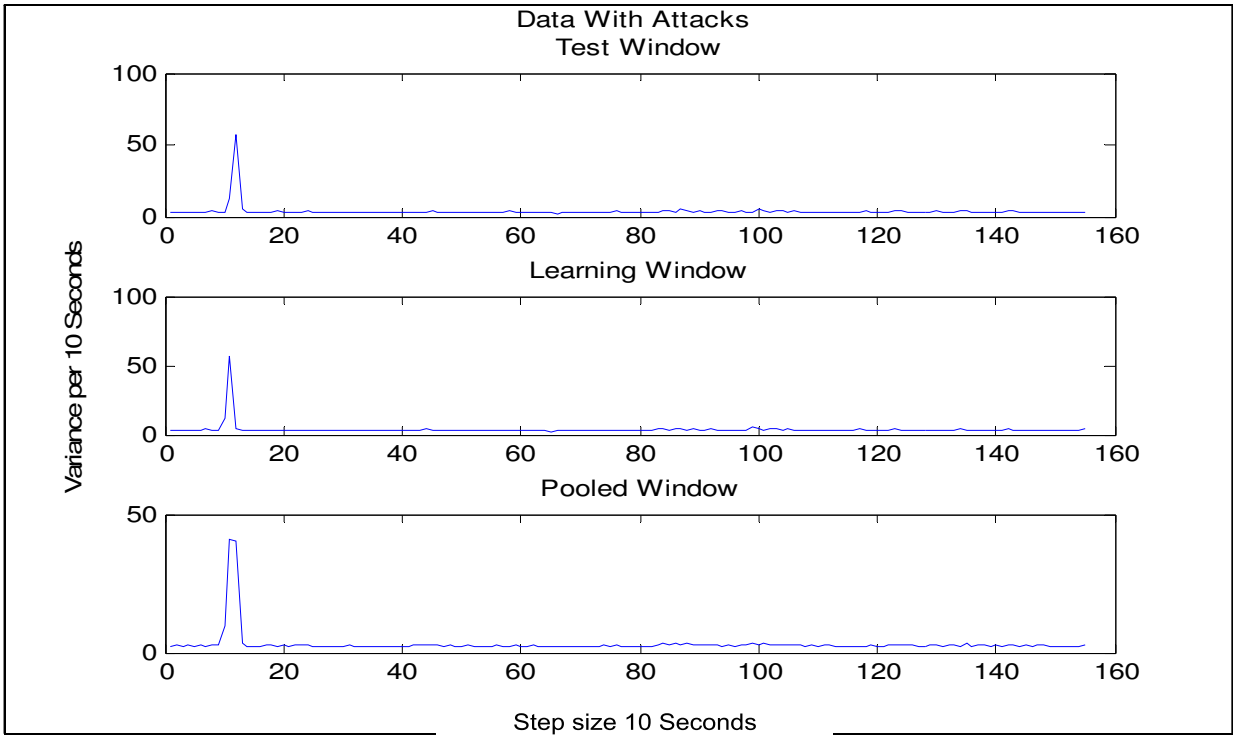

Figure 6.16. Variance of approximate coefficients of IPinreceives

Table 6.1. Variance and mean values of IP variables before and after wavelet analysis

Data with Attacks Dataset	Variance	Mean
IPinreceives	65,86	7,66
pdc1 (*A* with SVD, decision vector)	34,04	0,72
ax:level 1 App. Ceof. of IPinreceives	54,11	7,66
pdc1 (*A* with SVD, decision vector)	21,73	0,31
Attack Free data-1 dataset		
IPinreceives	2,59	4,17
pdc1 (*A* with SVD, decision vector)	6,25	0,31
ax:level 1 App. Ceof. of IPinreceives	2,25	4,18
pdc1 (*A* with SVD, decision vector)	2,86	0,13

As we understand from Table 6.1. we have eliminated the high variations from data. Variance values of *ax* (level-1 Approximate Coefficient of IPinreceives) are lower than IPinreceives. Decision variable "pdc1" variance is greater than variance of decision variance of *ax*. We can achieve these results via wavelet analysis it is because we are using level-1 approximate coefficients of IP variables. We are not using detailed coefficients so we are eliminating high variations from IP variables thus we reduce the false alarms without reducing the detection rate.

In Figure 6.15. and 6.16. we could not catch the second attack. Variance plots of IPindelivers and IPoutrequest are same fashion in other words we could not catch the second attack in other two variables either. Because of this we will not show all the variance plots.

We will show all three Likelihood Ratio η in Figure 6.17. and 6.18. In Figure 6.17. and 6.18. we can easily see the second attack but beside second attack there are other spikes which may yield false alarms in Figure 6.17.. From the Figure 6.17. and 6.18. we understand that Likelihood Ratio is affecting the decision positive way and it extracts the hidden information from input vectors which yields us a sensitive and deeper analysis method, but in some cases it is too deep analysis. We have added wavelet analysis in order to overcome false alarms, In Figure 6.18. we can see the differences from Figure 6.17. There is less background noise and noise is affecting the decision negative way. We have eliminated background noise effect from our decision vector in Figure 6.18. via wavelet analysis. In Figure 6.18. we can see clearly the second attack. We try to reduce the false

alarms without reducing the detection rate. From all above mentioned result we could say we have achieved this target. And also in variance and mean values of Data with Attack dataset and Attack Free Data-1 dataset we could see that we have reduced the variation in variables optimally so we have fewer false alarms with same detection rate.

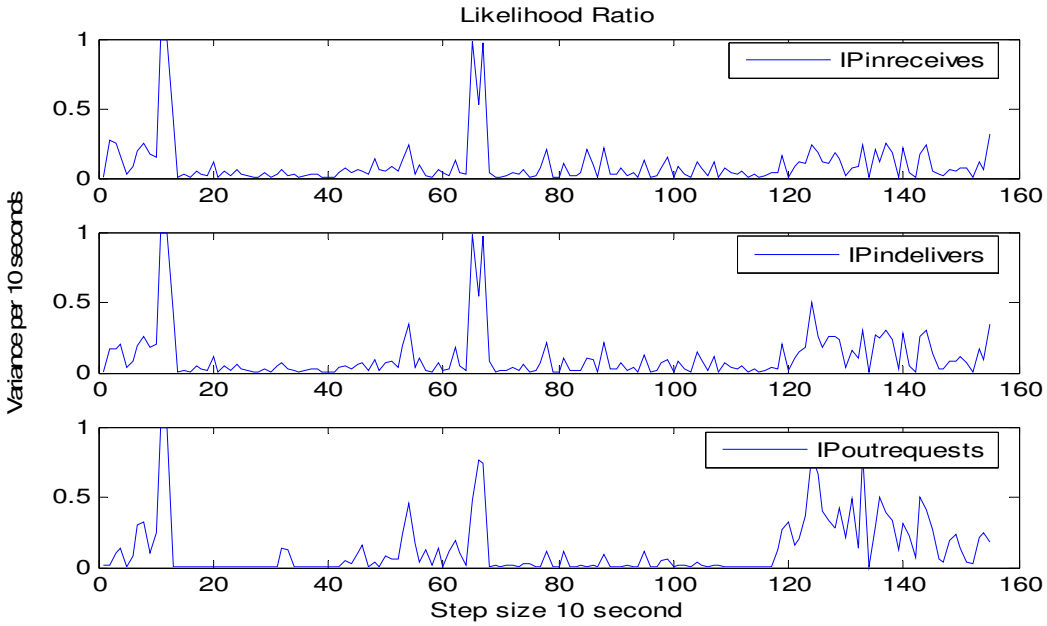

Figure 6.17. Likelihood ratios of IP variables

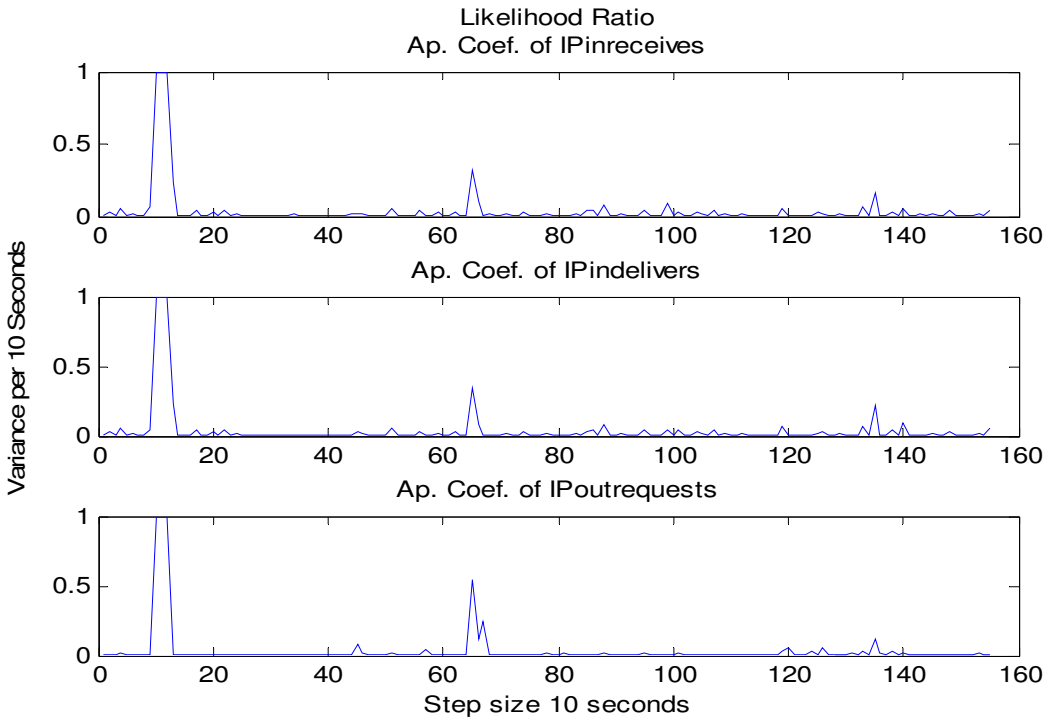

Figure 6.18. Likelihood ratios of level-1 approximate coefficients of IP variables

6.5.2. Pole-Zero Analysis on AR Model Coefficients

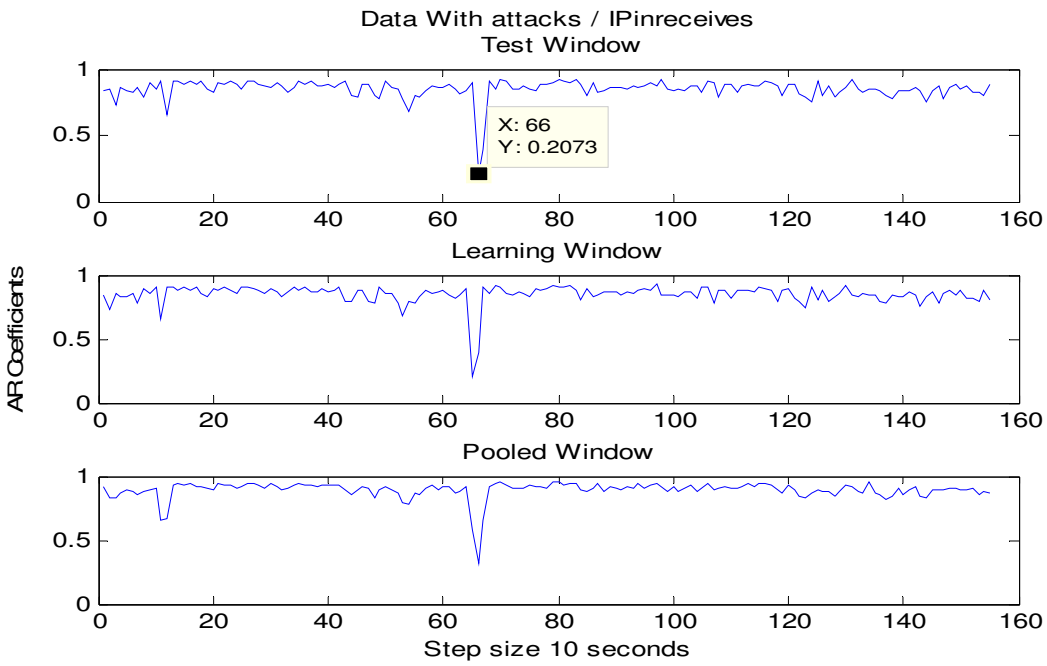

Figure 6.19. AR coefficients of data with attacks/IPinreceives

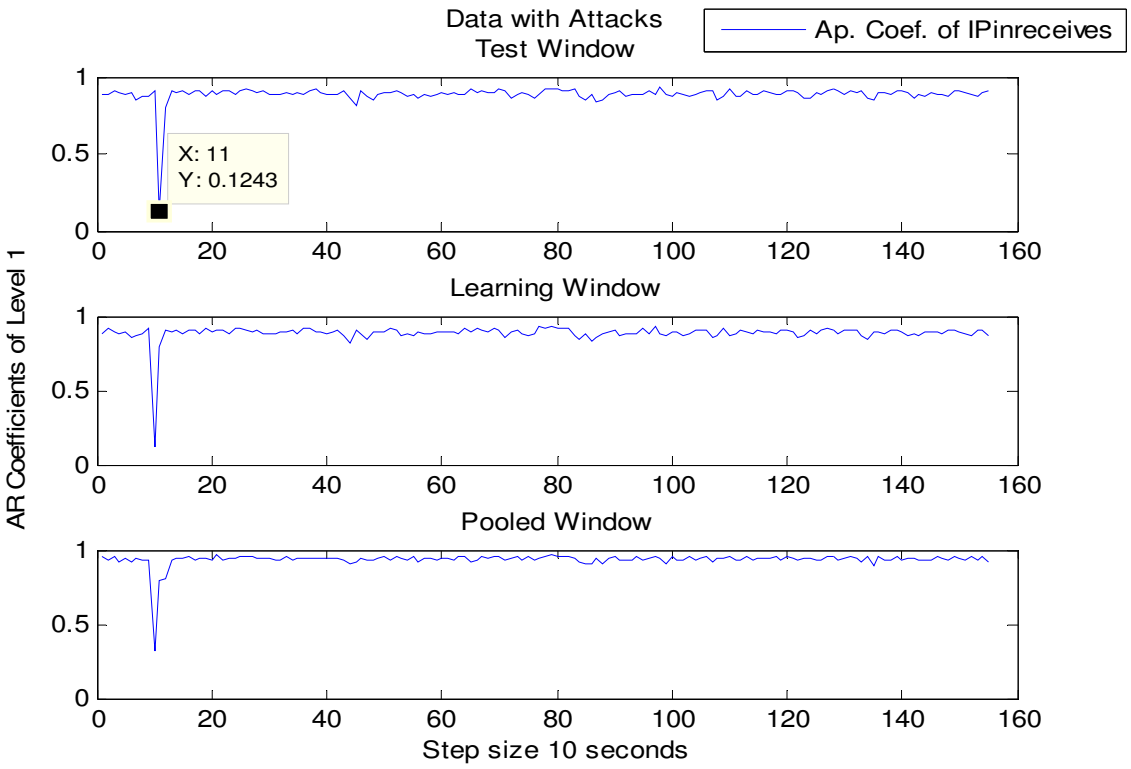

Figure 6.20. AR coefficients of data with attacks/level-1 approximate coefficients of IPinreceives

When we look at the Figure 6.19. and Figure 6.20. plots of AR coefficients we see there are two significant spikes which correspond to two significant attacks in our Data with Attacks dataset. AR model coefficients are between (0, 1) so we see that model is stable. We have added the pole-zero plots of these AR coefficient's roots in Figure 6.21. and Figure 6.22.. We can see that there are some significant roots are differentiating from others. We have marked some of them in Figure 6.19. and Figure 6.20. These marked roots are corresponding roots in Figure 6.21. and Figure 6.22. We could catch two attacks in Data with Attacks dataset in Likelihood Ratio plots and also in AR Coefficient's pole-zero plots.

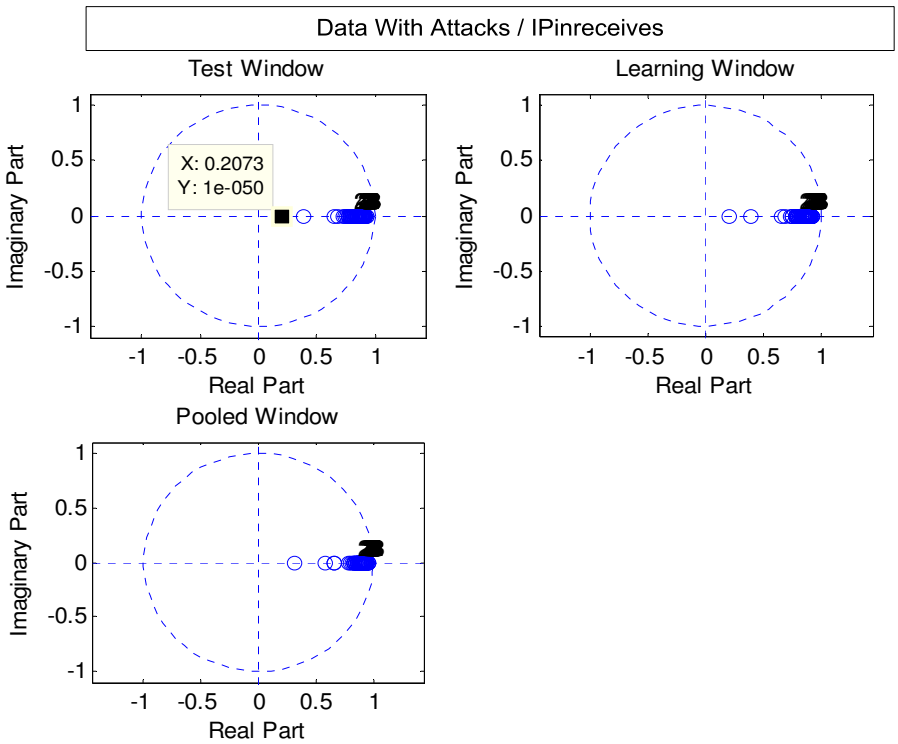

Figure 6.21. Data with attacks/IPinreceives AR coefficient's pole-zero plots

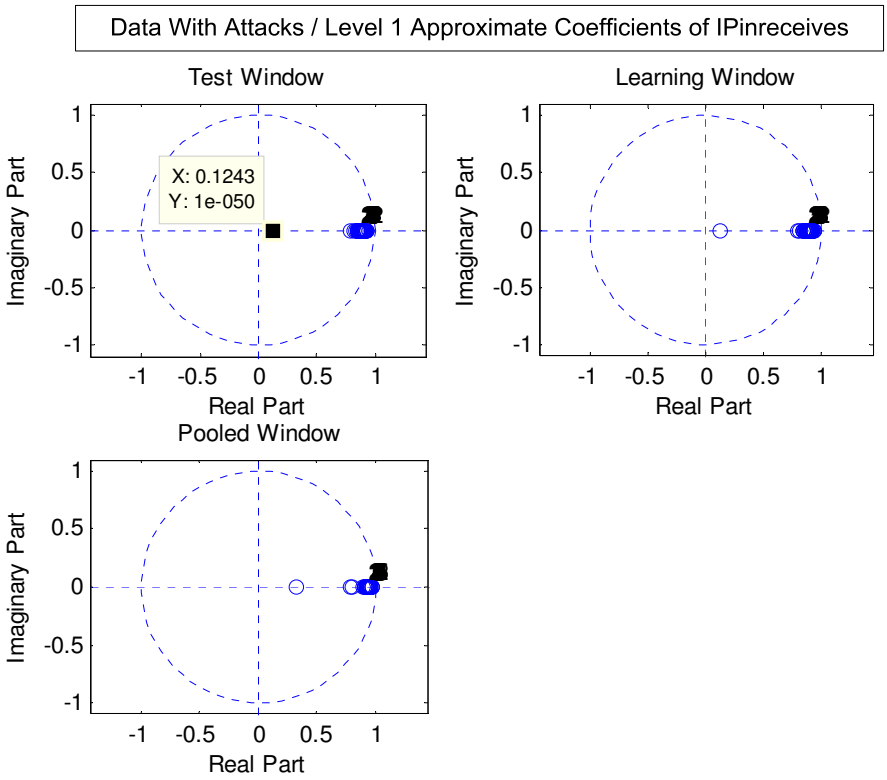

Figure 6.22. Data with attacks//level 1 approximate coefficients of IPinreceives AR coefficient's pole-zero plots

7. PERFORMANCE MEASUREMENTS

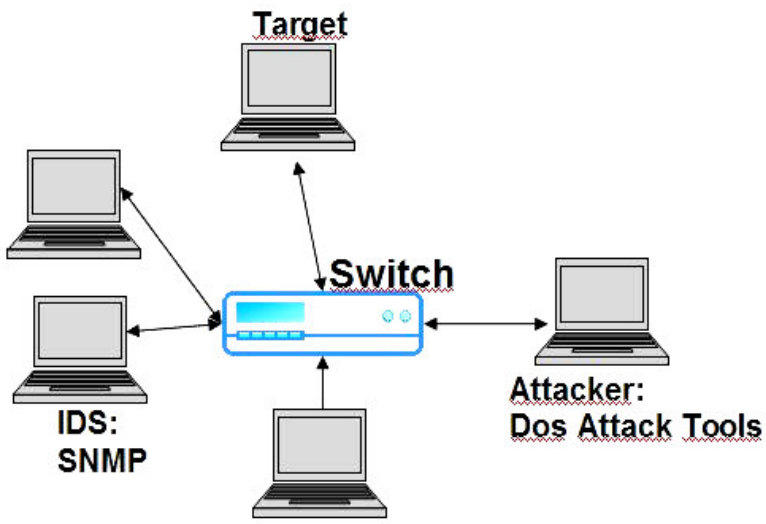

Figure 7.1. Topology for collecting SNMP data; dataset-1

We have compared Wavelet-Modulus Maxima, AR and Wavelet-AR models. We have used common performance parameters in [14]. *B* or *P(I)* is the base rate of data. The parameters we have been used: FP (False Positive, $P(A/I')$), TP (True Positive, $P(A/I)$), FN (False Negative, $P(A'/I)$), TN (True Negative, $P(A'/I')$), PPV (Positive Predictive Value or Bayesian Detection Rate, $P(I/A)$), NPV (Negative Predictive Value, $P(I'/A')$), and C_{ID} (Intrusion Detection Capability). C_{ID} is simply the ration of the mutual information between IDS input and output, and the entropy of the input[14].

$$PPV = P(I/A) = \frac{B(1-FN)}{B(1-FN)+(1-B)FP} \tag{7.1}$$

$$NPV = P(I'/A') = \frac{(1-B)(1-FP)}{(1-B)(1-FP)+B.FP} \tag{7.2}$$

$$C_{ID} = \frac{I(X;Y)}{H(X)} = \frac{(H(X)-H(X/Y))}{H(X)} \tag{7.3}$$

$$H(X) = -B\log B - (1-B)\log(1-B) \tag{7.4}$$

$$H(X/Y) = -B(1-FN)\log\frac{B(1-FN)}{B(1-FN)+(1-B)FP} - B.FN.\log\frac{B.FN}{B.FN+(1-B)(1-FP)} \tag{7.5}$$

$$-(1-B)(1-FP).\log\frac{(1-B)(1-FP)}{(1-B)(1-FP)+B.FN} - (1-B).FP.\log\frac{(1-B)FP}{(1-B).FP+B(1-FN)}$$

7.1. Artificial Data

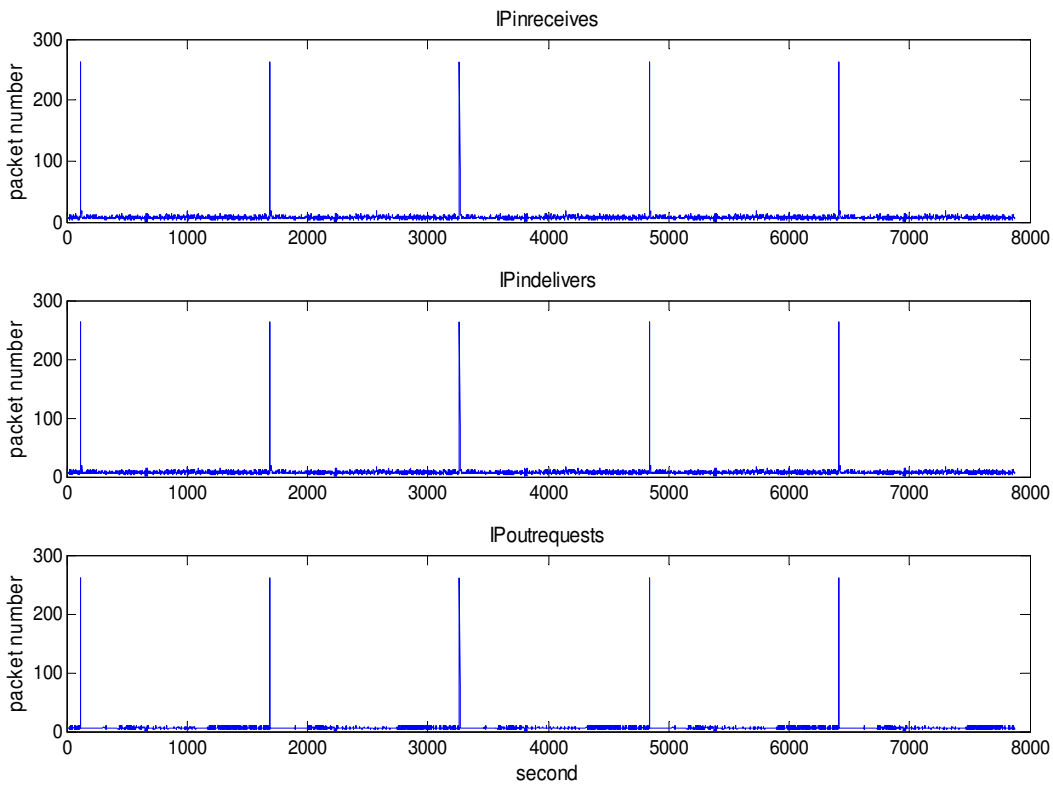

Figure 7.2. Artificial data; data with attacks dataset was added five times successively

7.1.1. AR Model Performance Measurements

We have 25 attacks units in data, 762 normal conditions units in data so $P(I) = 25/787$, $P(I) = 3.17$ per cent.

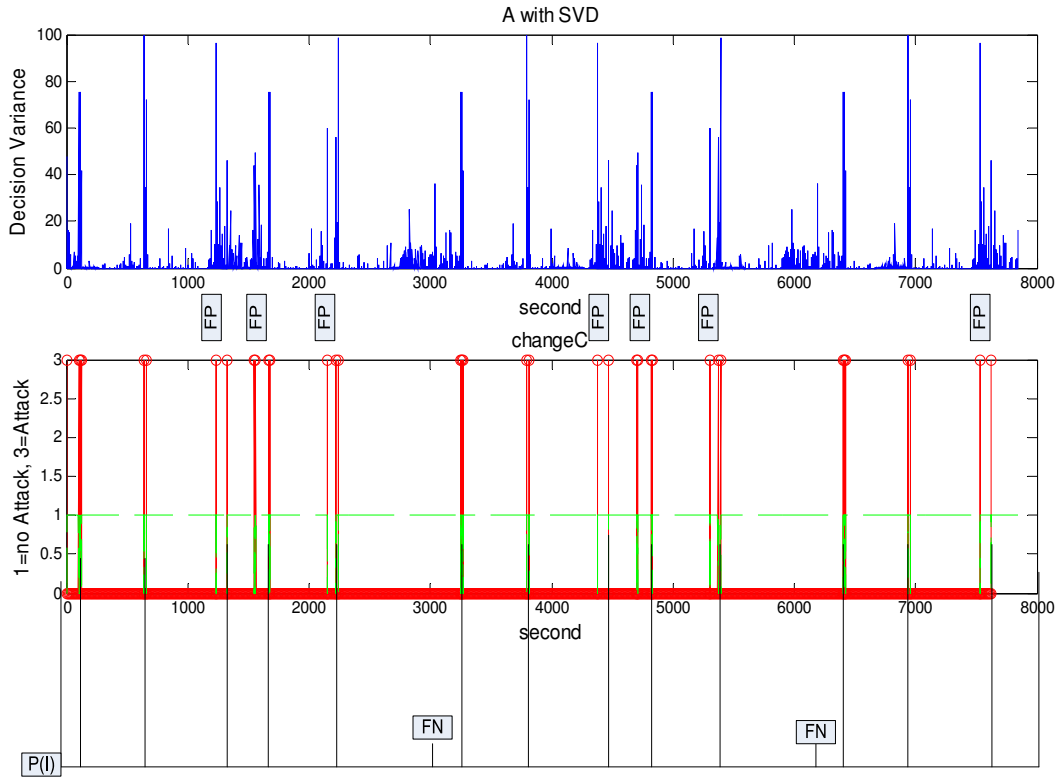

Figure 7.3. AR model decision variance, decision, and $P(I)$ real status of artificial data with shift length t=10sec

We have weighted attacks in the data. First attack was about 30 seconds in data it has three weighted constant, second and third attacks were about 10 seconds so they have one weighted constant. Data with attacks dataset contains five unit attack steps and we have added this data successively five times the total attack units are become 25. And normal unit steps are 762, total data length is 787 unit steps, actual data is 7870 seconds. We have done such a weighting because of the attack lengths are different. We have done 10 second to one unit transformation because of the AR model analysis structure which was taking 10 seconds or 10 samples and estimating the variance of this window so 10 second becomes one unit.

7.1.2. Wavelet-AR Model Performance Measurements

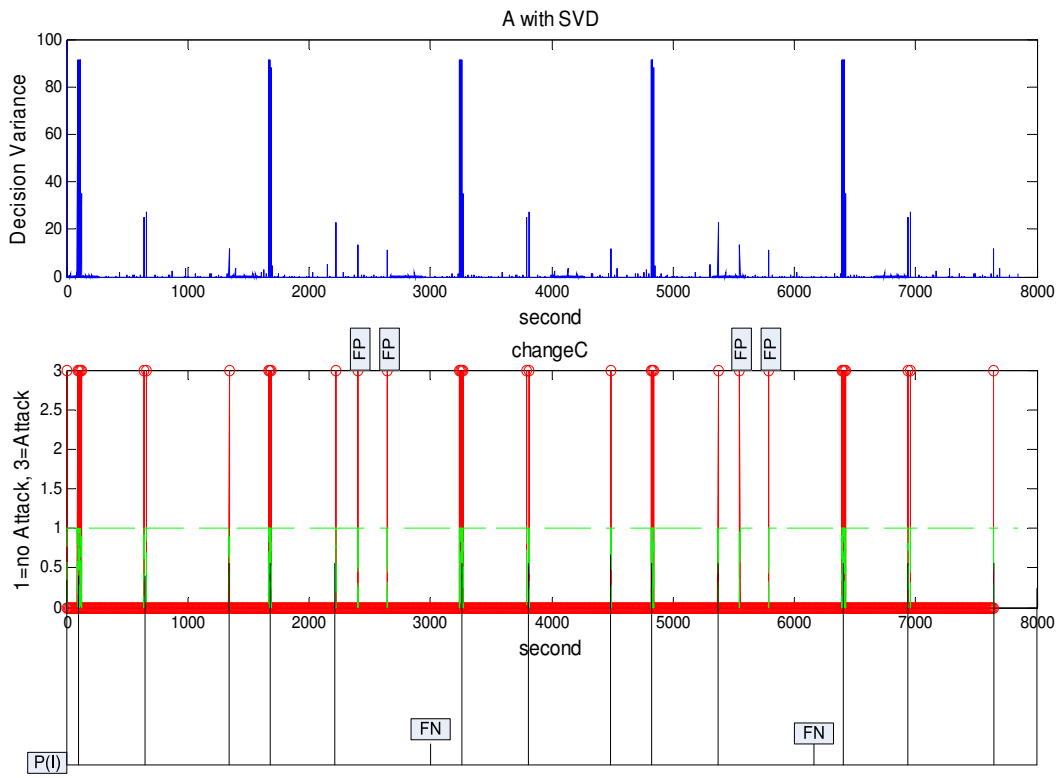

Figure 7.4. Wavelet- AR model decision variance, decision, and $P(I)$ real status of artificial data with shift length t=10sec

Table 7.1. Wavelet-Modulus Maxima, AR and Wavelet-AR models performance measurements of artificial data with t=10sec

Model	Alarms	No-Alarms	FP	TN	TP	FN	PPV	NPV	Cid
Wavelet-Modulus Maxima	15	772	0,0000	1,0000	0,6000	0,4000	1,0000	0.98707	0.51824
AR Model (SVD)	32	755	0,0118	0,9882	0,9200	0,0800	0.7185	0.99736	0.70334
Wavelet- AR model (SVD)	27	760	0,0052	0,9948	0,9200	0,0800	0.85277	0.99737	0.77317

From Table 7.1. we could see that Wavelet-Modulus Maxima has poor detection rate. We could see that AR model and Wavelet-AR model have same TP and FN rates, but AR model's FP rate is higher than Wavelet-AR model. And Wavelet-AR model has better TN rate than AR model. When we look at the NPV they are same, but Wavelet-AR model has greater PPV than AR model. These measurements show that Wavelet-AR model is

better than AR model regarding FP and PPV this means Wavelet-AR IDS is better than AR-IDS. Wavelet-AR model's C_{ID} is greater than AR model this also verifies that Wavelet-AR IDS is better than AR-IDS.

We have performed test on new LLR (η_L) which is proposed in [32] with our models. Results show that new LLR has not good detection rate to use as an IDS. We have shown performance measurements on Table 7.2. As shown in Figure 7.5. new LLR has lots of False alarms.

Table 7.2. AR and Wavelet-AR models performance measurements of artificial data with new LLR (η_L)

Model	Alarms	No-Alarms	FP	TN	TP	FN	PPV	NPV	Cid
AR Model (SVD)	18	769	0,0000	1,0000	0,7200	0,2800	1,0000	0.99092	0.64037
Wavelet-AR model (SVD)	24	763	0,0079	0,9921	0,7200	0,2800	0.74898	0.99085	0.51875

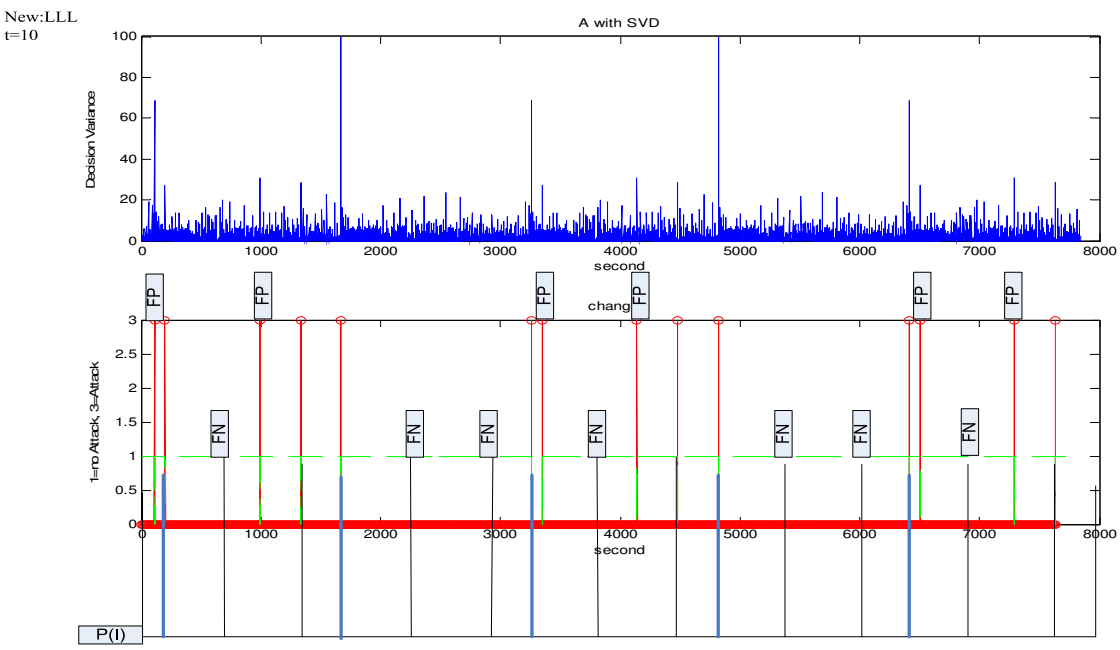

Figure 7.5. Wavelet- AR model decision variance, decision, and $P(I)$ real status of artificial data with new LLR and shift length t=10sec

We have used *t* as a shift window. Firstly, *t*=10 seconds this also the length of learning and test windows. We have found that if an attack does not fit with test widow like shown in Figure 7.6. We can not detect it.

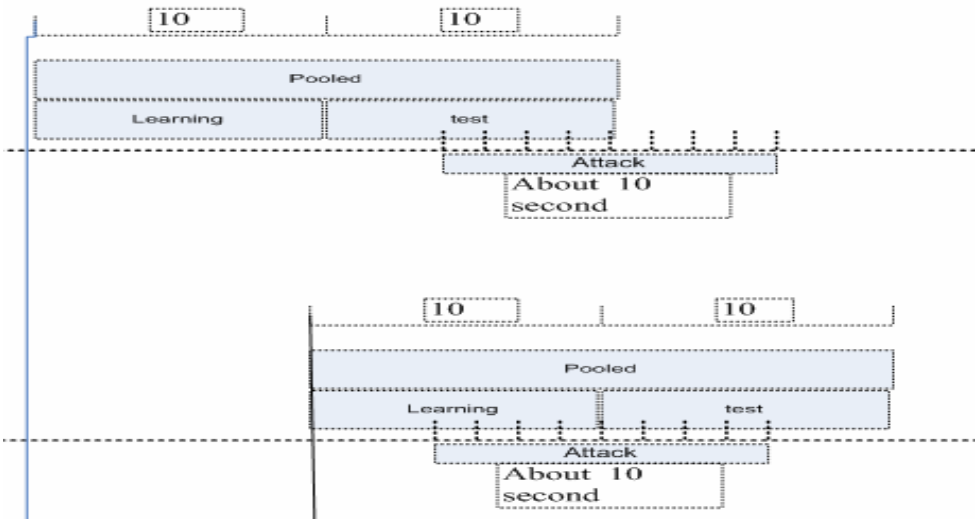

Figure 7.6. Miss attack representation

We have used *t*=5sec. while learning and test window were equal to 10sec. With this way we could detect the some missed attacks, but beside this false alarms were increased also. Wavelet-AR model showed acceptable False Positive rate and PPV. All the results are shown in Table 7.3.

Table 7.3. AR and Wavelet-AR models performance measurements of artificial data with *t*=5sec. and *t*=10sec

Model	**Alarms**	**No-Alarms**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	t(sec)
AR Model (SVD)	32	755	0,0118	0,9882	0,9200	0,0800	0.7185	0.99736	0.70334	10
Wavelet-AR model (SVD)	27	760	0,0052	0,9948	0,9200	0,0800	0.85277	0.99737	0.77317	10
AR Model (SVD)	47	740	0,0289	0,9711	1,0000	0,0000	0.53113	1,0000	0.70659	5
Wavelet-AR model (SVD)	39	748	0,0184	0,9816	1,0000	0,0000	0.64019	1,0000	0.76992	5

Performance measurements have showed that Wavelet-AR model with shift length *t*=5sec has the best results and highest C_{ID}. In other words shift length *t*=5sec has the best detection rate with acceptable False Positive rate. We have used firstly proposed likelihood ratio which has better detection rate.

As shown in Figure 7.7. there is not any missed attack, there are a few false alarms. The Wavelet-AR model has fewer false alarms than AR model.

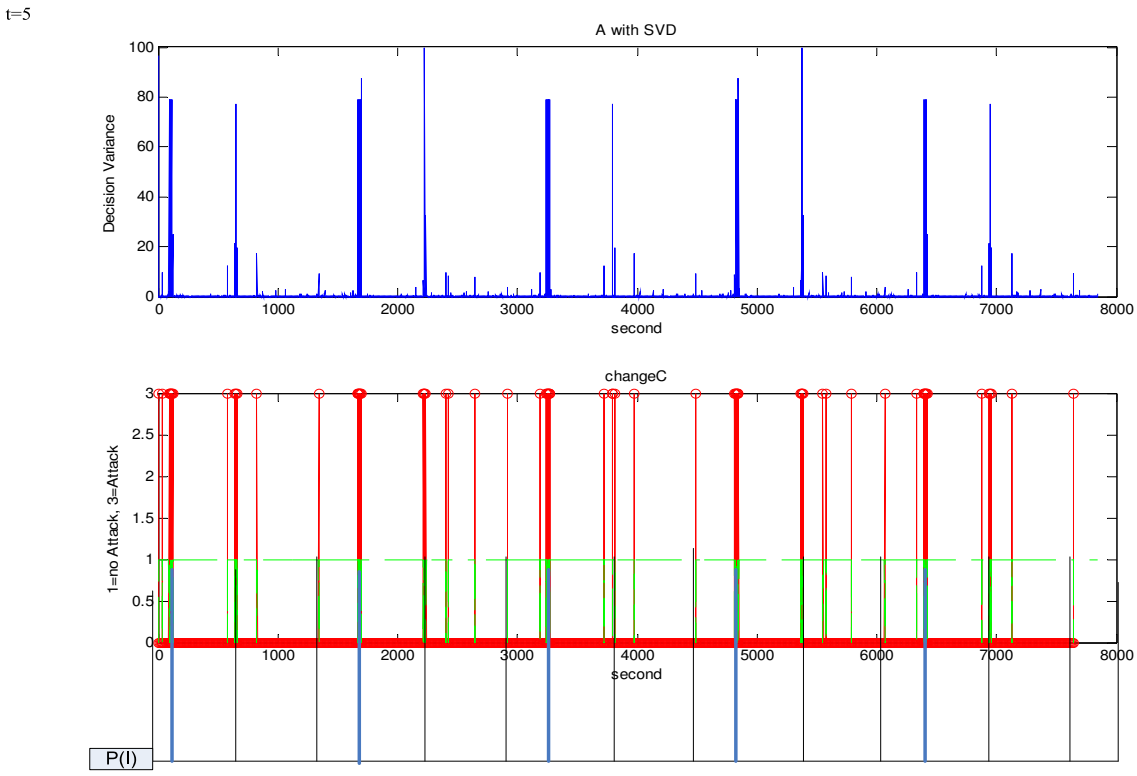

Figure 7.7. Wavelet- AR model decision variance, decision, and $P(I)$ real status of artificial data with first η and shift length *t*=5sec

7.1.3. Wavelet-Modulus Maxima Model Performance Measurements

Figure 7.8. shows that Wavelet-Modulus Maxima has low detection rate. This method is not good for Ethernet traffic data.

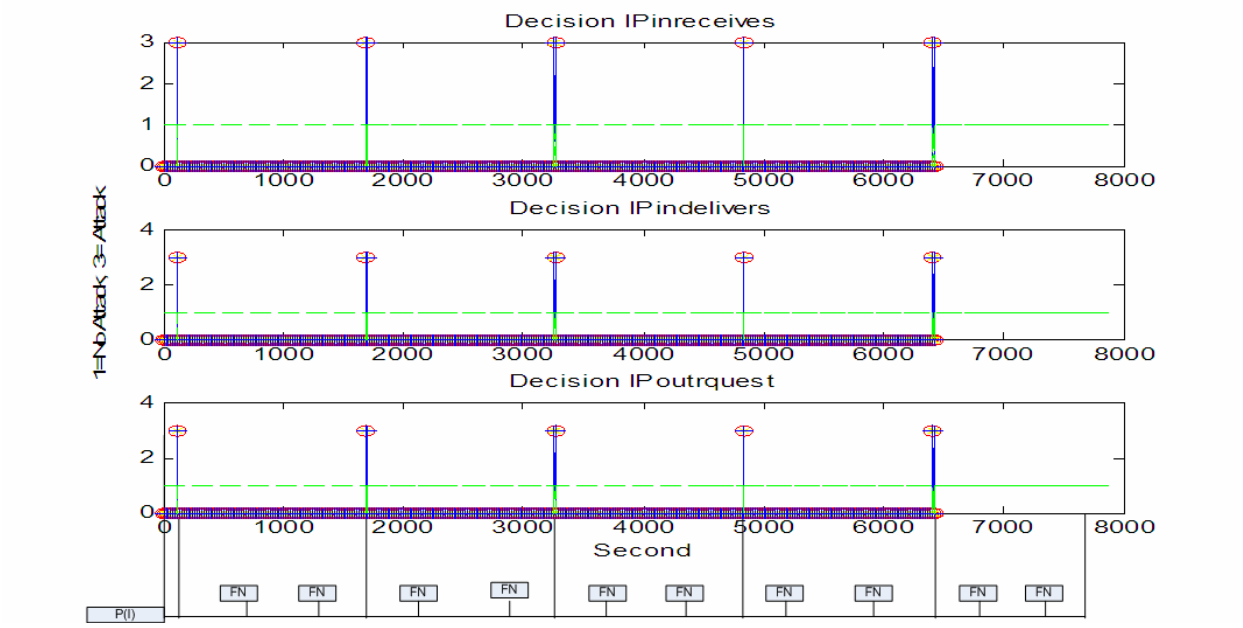

Figure 7.8. Wavelet- Modulus Maxima model, decision, and $P(I)$ real status of data

7.2. CASE-1 ADSL Modem Working as a Switch Mode

These data were captured in dataset-2. The network was represented in Figure 5.12. was used for data capturing. We have collected nine different data and their names are test2_02, test3_02, test1_03, test2_03, test4_03, test6_03, test7_03, test8_03, and test9_03. Each test is includes 42 minutes data and each test includes 500 steps of data. The data includes the variables IPinreceives, IPindelivers, IPoutrequests, and Ifinoctets. The important property of this dataset is ADSL modem was working as a switch.

7.2.1. Test2_02 Data

Test2_02 data include six attacks as we can see from Figure 7.9. and attacks are brute force and port scan. All these attacks have been done to ADSL.

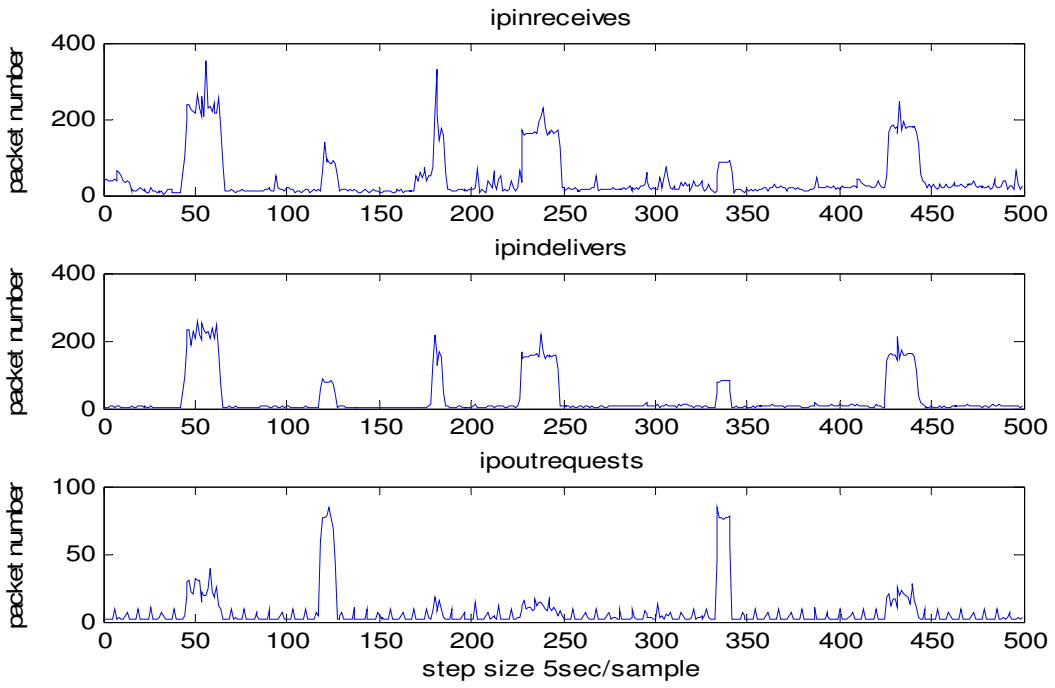

Figure 7.9. Test2_02 data includes IP variables

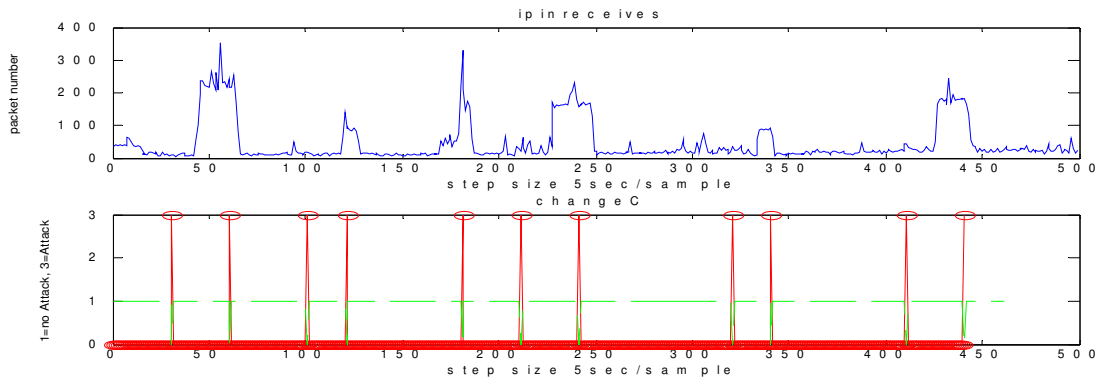

Figure 7.10. Test2_02, Wavelet-AR model decision vector and IPinreceives variables, t=10

As we can see from Figure 7.10. Wavelet-AR model can detect all the attacks towards to ADSL modem which comes from internal network. And also as we can see from Table 7.4. we have high detection rates at all models, but we should not forget that false alarm ratio is also an important parameter. From Table 7.4. AR model has high false

alarm rate so it makes this model less reliable than Wavelet-AR model. *t* is the shift window in Wavelet-AR model.

Table 7.4. Test2_02 performance measurements

Test2_02	Base Rate	Alarms	No-Alarms	FP	TN	TP	FN	PPV	NPV	Cid	t(sec)
Wavelet-AR model (SVD)	38,00%	21	29	0,065	0,935	1,000	0,000	0,900	1,000	0,801	10
AR Model (SVD)	38,00%	35	15	0,516	0,483	1,000	0,000	0,540	1,000	0,273	10
Wavelet-AR model (SVD)	29,00%	41	59	0,169	0,830	1,000	0,000	0,707	1,000	0,588	5
AR Model (SVD)	29,00%	61	39	0,450	0,540	1,000	0,000	0,470	1,000	0,299	5

7.2.2. Test4_03 Data

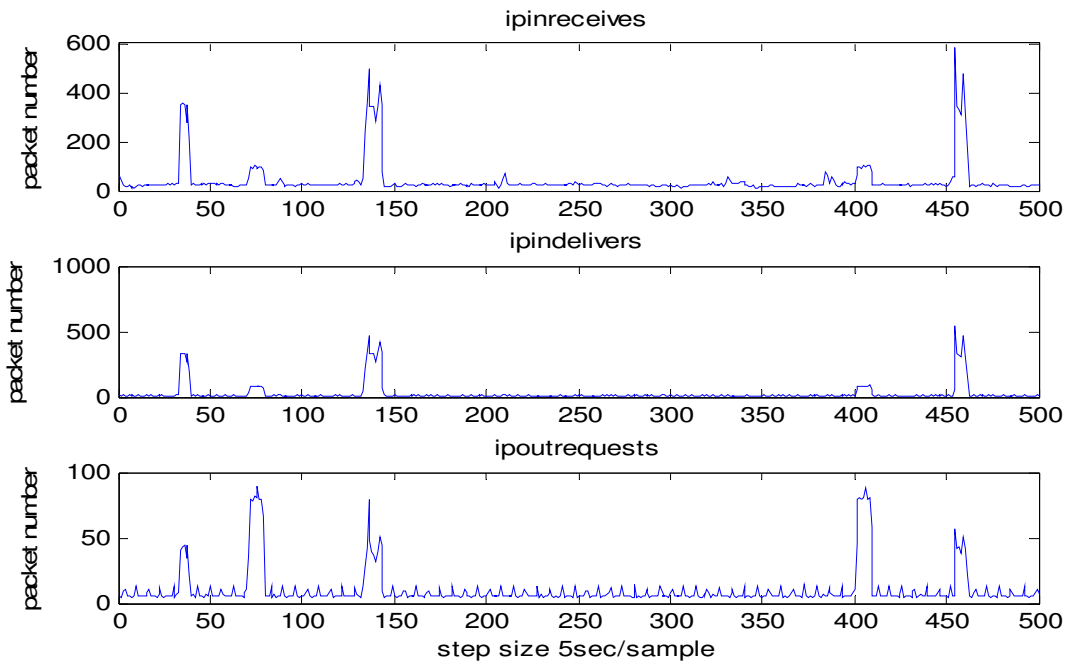

Figure 7.11. Test4_03 data includes IP variables

Test4_03 data include nine attacks but we can only see five of them in the Figure 7.11., the included attacks are brute force and port scan. The attacks can be easily seen from Figure 7.11. all towards to ADSL modem and there are four attacks to PC2 and PC3 in Dataset-2 Network.

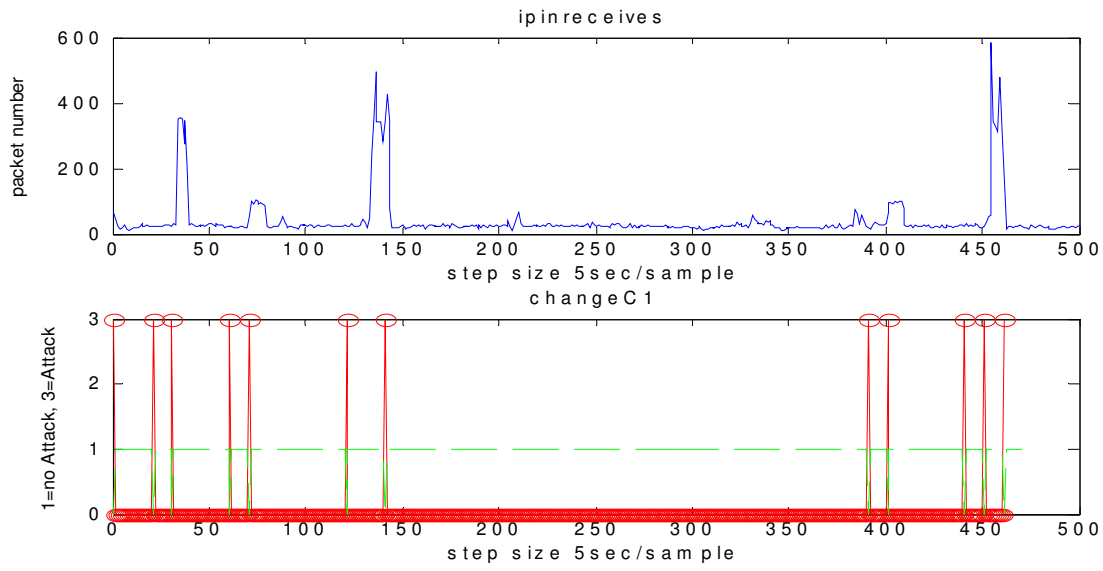

Figure 7.12. Test4_03 data, Wavelet-AR model decision vector and IPinreceives variables, t=10, 3 input

As we can see from Figure 7.12. Wavelet-AR model can detect all the attacks towards to ADSL modem which comes from internal network, but it can not detect attacks towards to PC2 and PC3. We have added the variable Ifinoctet to Wavelet-AR model. Interface variable includes all the data regarding all physical ports and regardless of Layers and Protocols. Because of we are working switch mode and also we are collecting the variables regarding IP Layer we can not see the peaks related attacks to PC2 and PC3 in our traffic flow.

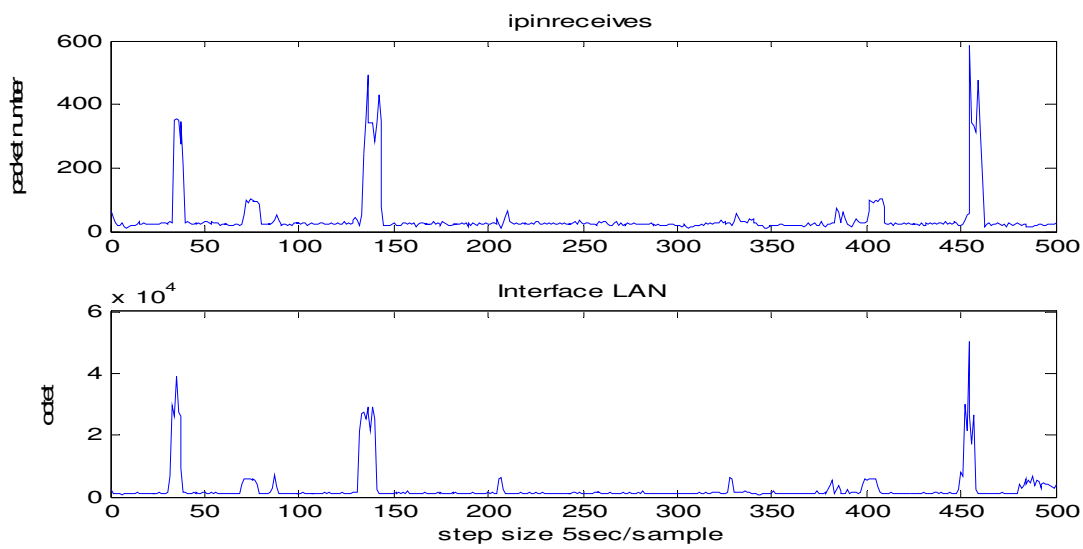

Figure 7.13. Test4_03 data, interface LAN variable has been added

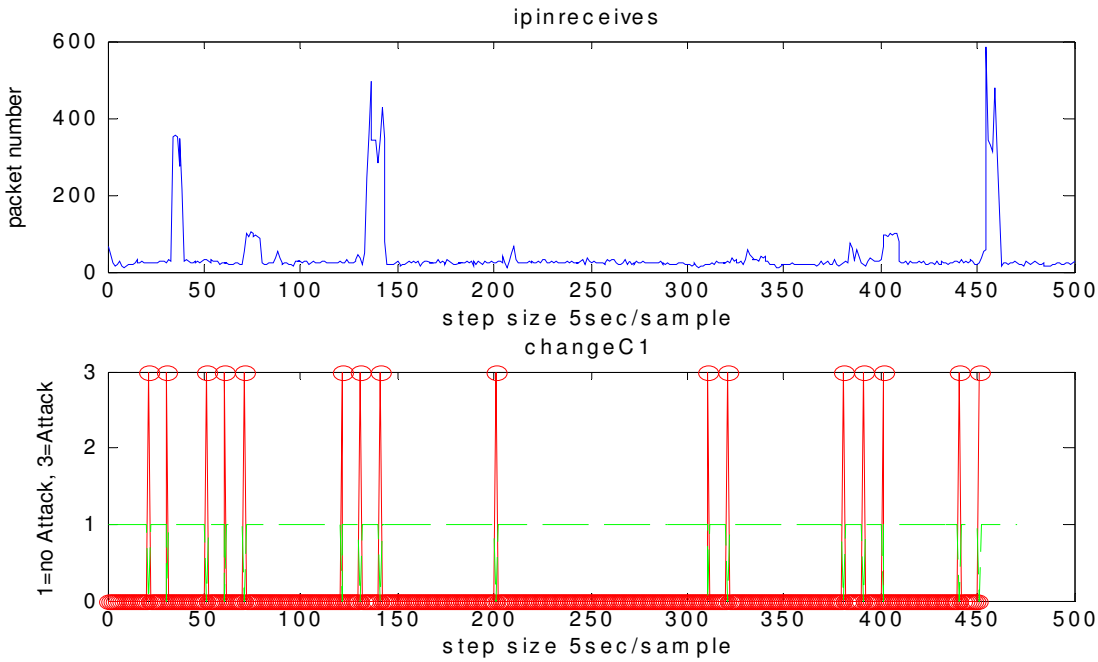

Figure 7.14. Test4_03 data, Wavelet-AR model decision vector and IPinreceives variables, t=10, 4 input

In Figure 7.13. we can see some small peaks that corresponds to the brute force attacks to PC2 and PC3. As we can see from Figure 7.14. Wavelet-AR model can detect two of the hidden attacks. We could say that, the interface LAN variable increase the detection rate when ADSL modem was working in switch mode.

Table 7.5. Test4_03 performance measurements with three IP variables

Test4_03, 3 input	**Base Rate**	**Alarms**	**No-Alarms**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	t(sec)
Wavelet-AR model (SVD)	40,00%	13	37	0,033	0,966	0,600	0,400	0,923	0,783	0,321	10
AR Model (SVD)	40,00%	12	38	0,000	1,000	0,600	0,400	1,000	0,789	0,418	10
Wavelet-AR model (SVD)	37,00%	29	71	0,063	0,936	0,675	0,324	0,862	0,830	0,333	5
AR Model (SVD)	37,00%	27	73	0,031	0,968	0,675	0,324	0,925	0,835	0,396	5

Table 7.6. Test4_03 performance measurements, LAN variable was added

Test4_03, 4 input	Base Rate	Alarms	No-Alarms	FP	TN	TP	FN	PPV	NPV	Cid	t(sec)
Wavelet-AR model (SVD)	40,00%	18	32	0,066	0,933	0,800	0,200	0,880	0,875	0,455	10
AR Model (SVD)	40,00%	14	36	0,066	0,933	0,600	0,400	0,857	0,777	0,262	10
Wavelet-AR model (SVD)	37,00%	39	61	0,126	0,873	0,837	0,162	0,794	0,901	0,402	5
AR Model (SVD)	37,00%	31	69	0,048	0,952	0,756	0,243	0,903	0,869	0,444	5

When we look at the values of Table 7.5. and Table 7.6. we can see that Table 7.6. has better True Positive values than Table 7.5. This detection rate increase comes from variable LAN. We can fairly say that LAN variable has increased the Wavelet-AR and AR models detection rate when ADSL modem was working in switch mode.

7.2.3. Test8_03 Data

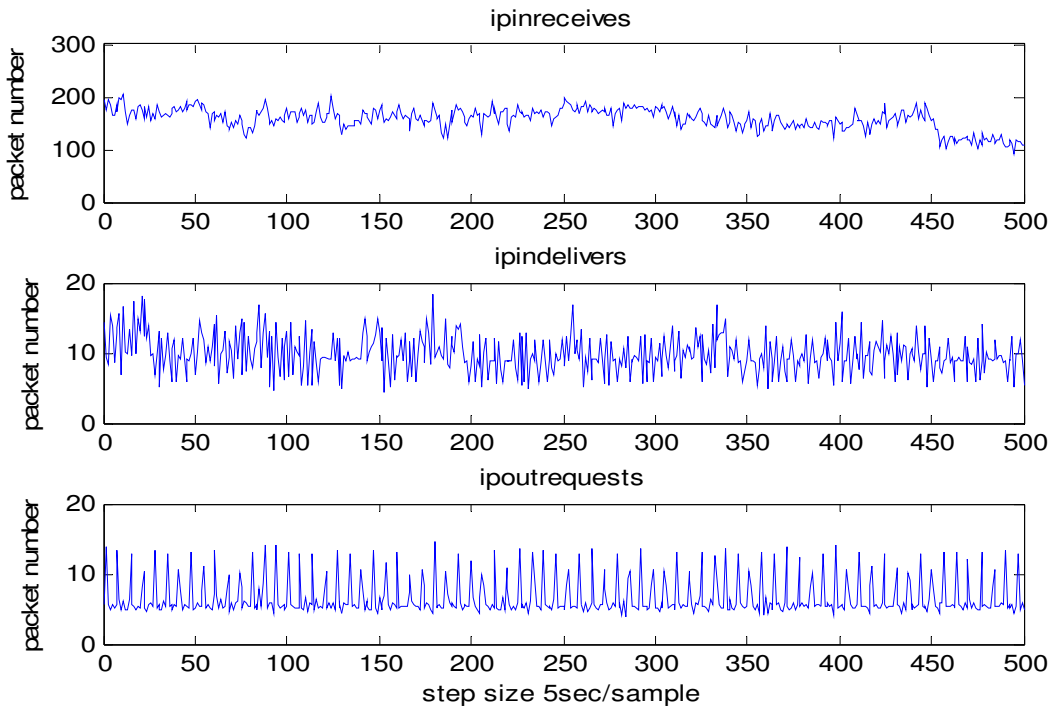

Figure 7.15. Test8_03 data includes IP variables

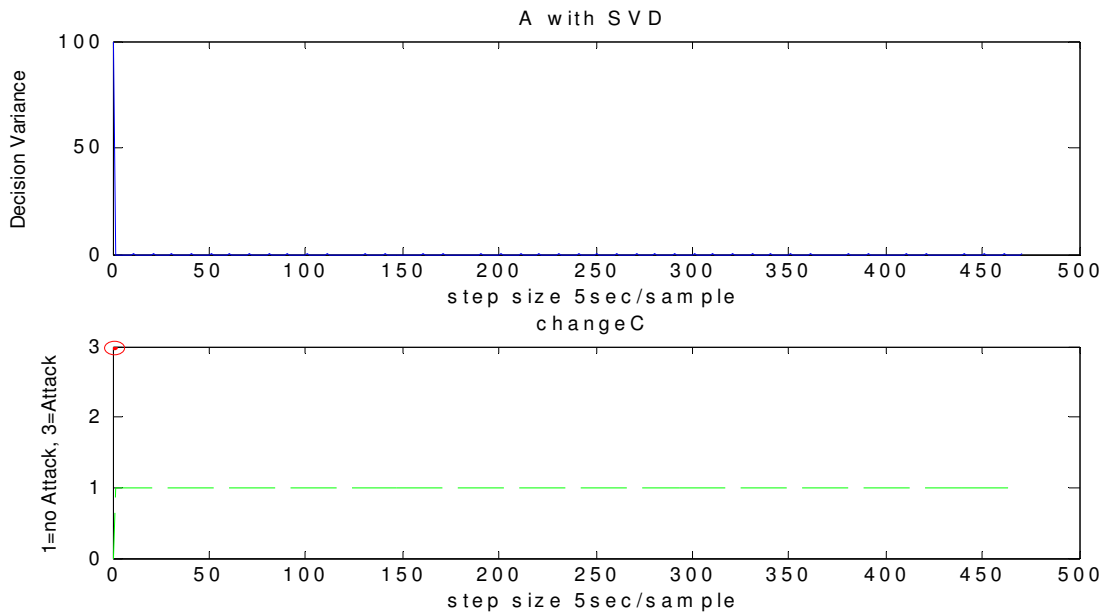

Figure 7.16. Test8_03 data, Wavelet-AR model decision vector and IPinreceives variables, t=10, 3 input

We have performed Wavelet-AR model with Test8_03 attack free data. From Figure 16. we can see that Wavelet-AR model has no false alarms, it means that Wavelet-AR models is reliable and robust with attack free data.

7.2.4. Wavelet-AR and AR Model Tables

The *t* is the shift window in Wavelet-AR and AR models. While we are changing shift window we do not change Learning and Test windows length.

Table 7.7. Test_All_0203 Wavelet-AR model four input performance measurements

t=10	**I**	**Normal**		**Alarm**	**Normal**	**Detected**		**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**
test2_02	19	31		21	29	19		0,065	0,935	1,000	0,000	0,900	1,000	0,801
test3_02	19	31		22	28	19		0,096	0,903	1,000	0,000	0,863	1,000	0,736
test1_03	21	29		23	28	21		0,032	0,967	1,000	0,000	0,950	1,000	0,880
test2_03	20	30		22	28	20		0,066	0,933	1,000	0,000	0,909	1,000	0,800
test4_03	20	30		18	32	16		0,066	0,933	0,800	0,200	0,880	0,875	0,455
test6_03	20	30		13	37	12		0,033	0,966	0,600	0,400	0,932	0,783	0,321
test7_03	20	30		19	31	18		0,033	0,966	0,900	0,100	0,947	0,935	0,663
Toplam	**139**	**211**		**138**	**213**	**125**		**0,057**	**0,943**	**0,899**	**0,101**	**0,912**	**0,934**	**0,607**
Base Rate	**0,397**													

Table 7.8. Test_All_0203 AR model four input performance measurements

t=10	I	Normal		Alarm	Normal	Detected		FP	TN	TP	FN	PPV	NPV	Cid
test2_02	19	31		35	15	19		0,516	0,483	1,000	0,000	0,540	1,000	0,273
test3_02	19	31		22	28	19		0,096	0,903	1,000	0,000	0,863	1,000	0,736
test1_03	21	29		25	25	21		0,161	0,838	1,000	0,000	0,791	1,000	0,630
test2_03	20	30		21	29	20		0,033	0,966	1,000	0,000	0,952	1,000	0,880
test4_03	20	30		14	36	12		0,066	0,933	0,600	0,400	0,857	0,777	0,262
test6_03	20	30		15	35	14		0,033	0,966	0,700	0,300	0,933	0,828	0,414
test7_03	20	30		25	25	20		0,166	0,833	1,000	0,000	0,800	1,000	0,628
Total:	**139**	**211**		**157**	**193**	**125**		**0,152**	**0,848**	**0,899**	**0,101**	**0,796**	**0,927**	**0,449**
Base Rate	0,397													

7.2.5. Comments on Dataset-2

Brute Force Attacks towards to PCs show traffic changes on IPinreceives and LAN traffic. Brute Force Attacks towards to ADSL Modem show traffic changes on IPindelivers, IPinreceives, IPoutrequests, and LAN traffic.

Wavelet-AR model can detect all the attacks towards to ADSL modem which comes from internal network, but it can not detect attacks towards to PC2 and PC3. We have added the variable Ifinoctet to Wavelet-AR model. The analysis done with four inputs (IP variables and Ifinoctet LAN variable) has better True Positive values than the analysis done with three inputs (only IP variables). This detection rate increase comes with the help of variable LAN. We can fairly say that LAN variable has increased the model detection rate when ADSL modem was working in switch mode.

Table 7.9. Test_All_0203 W-AR versus AR comparison table

						Partitioned									
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	Total:	139	211	130	220	119	0,052	0,948	0,856	0,144	0,915	0,909	0,555	10
4 in	W-AR	Total:	139	211	138	213	125	0,057	0,943	0,899	0,101	0,912	0,934	0,607	10
3 in	AR	Total:	139	211	145	205	121	0,114	0,886	0,871	0,129	0,834	0,912	0,464	10
4 in	AR	Total:	139	211	157	193	125	0,152	0,848	0,899	0,101	0,796	0,927	0,449	10
						Combined									
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	Total:	139	211	124	226	119	0,023	0,976	0,856	0,143	0,956	0,911	0,623	10
4 in	W-AR	Total:	139	211	125	225	119	0,028	0,975	0,856	0,143	0,952	0,110	0,610	10
3 in	AR	Total:	139	211	124	226	119	0,023	0,976	0,856	0,143	0,956	0,911	0,623	10
4 in	AR	Total:	139	211	120	230	113	0,033	0,966	0,812	0,187	0,941	0,886	0,541	10

7.3. CASE-2 ADSL Modem Working as Router and Switch Modes

These data were captured in dataset-3. The network was represented in Figure 5.13. and Figure 5.14. were used for data capturing. We have collected eight different data, their names are test1_21, test2_21, test3_21, and test4_21 we have called these four tests Part_1. Others are test5_21, test6_21, test7_21, and test8_21 we have called these four tests Part_2. Each test is includes 42 minutes data and each test includes 500 steps of data. The data includes the variables IPinreceives, IPindelivers, IPoutrequests, and Ifinoctets. The important property of this dataset is ADSL modem was working like both Router and Switch. ADSL modem was working as a Router between Blue-Net and Red-Net. ADSL modem was working as a Switch in Red-Net same as in Blue-Net. We have called this setup Dataset-3.

7.3.1. Test1_21 Data

Test1_21 data include six attacks; the included attacks are brute force and port scan.

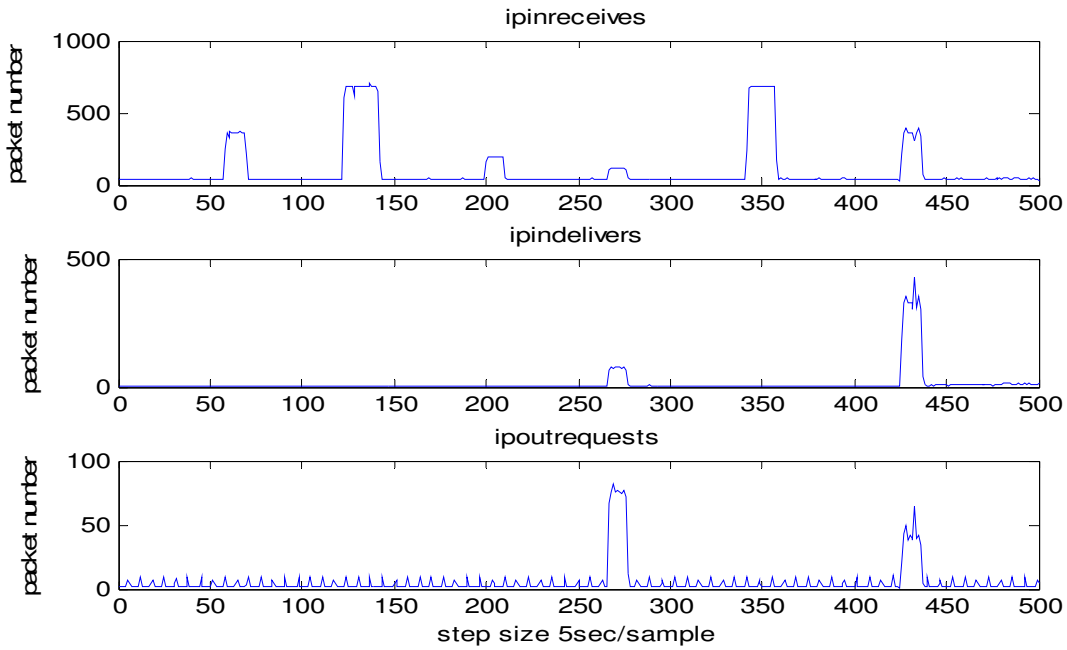

Figure 7.17. Test1_21 data IP variables

The attacks can be seen from IPindelivers and IPoutrequests all towards to ADSL modem and there are four attacks to PC2 and PC3 in Test1_21. These four attacks can be

seen from IPinrecieves variables only this shows us the importance of IPinreceives variables.

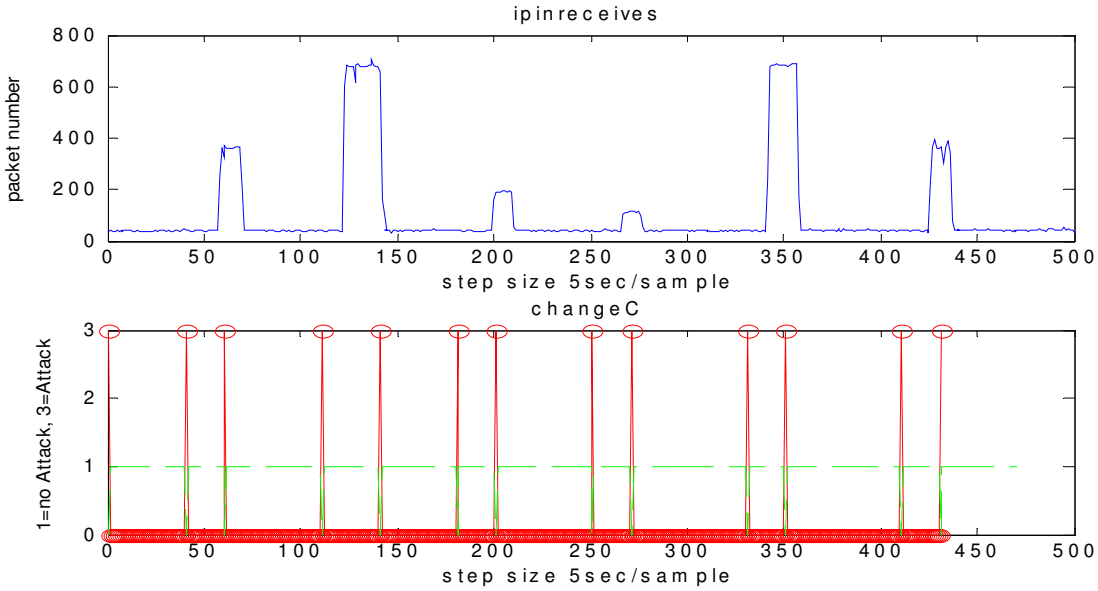

Figure 7.18. Test1_21 Wavelet-AR model decision vector and IPinreceives variables, t=10, 3 input

As we can see from Figure 7.18. Wavelet-AR model can detect all the attacks towards to ADSL modem and PCs. And also as we can see from Table 7.10. we have high detection rates at all models. Table 7.10. also shows that Wavelet-AR model can detects all the attacks when ADSL modem is in Router mode. *t* is the shift window.

Table 7.10. Performance measurements of Test1_21

Test1_21, 3 input	**Base Rate**	**Alarms**	**No-Alarms**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	t(sec)
Wavelet-AR model (SVD)	38,00%	19	31	0,000	1,000	1,000	0,000	1,000	1,000	0,999	10
AR Model (SVD)	38,00%	23	27	0,129	0,870	1,000	0,000	0,826	1,000	0,679	10
Wavelet-AR model (SVD)	35,00%	35	65	0,000	1,000	1,000	0,000	1,000	1,000	0,999	5
AR Model (SVD)	35,00%	36	64	0,015	0,984	1,000	0,000	0,972	1,000	0,929	5

7.3.2. Test4_21 Data

Test4_21 data include six attacks, the included attacks are brute force and port scan.

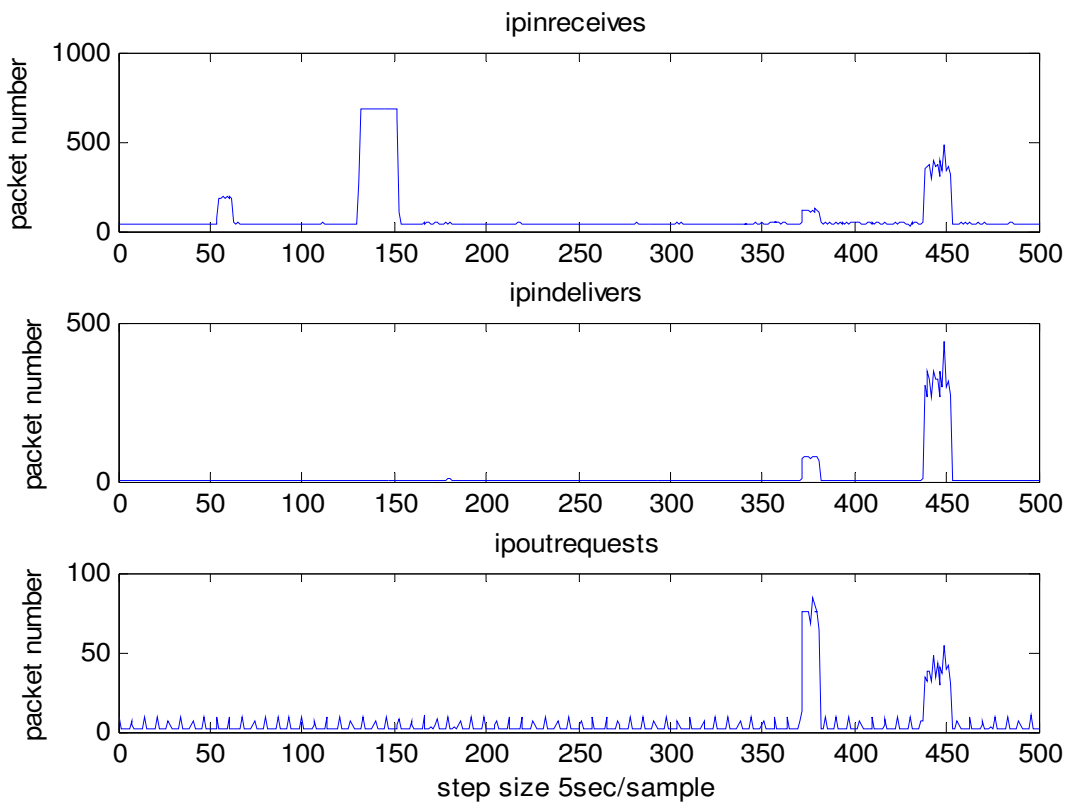

Figure 7.19. Test4_21 data IP variables

The attacks can be seen from IPindelivers and IPoutrequests all towards to ADSL modem and there are four attacks to PC2 and PC3 in Test4_21, Test4_21 was collected in Dataset-3 Network-2. Two of them can be seen from IPinrecieves variables only. In this setup ADSL modem was working like Router and Switch so we can not see the attacks in the same network. We can see the attacks between different networks.

As we can see from Figure 7.19. Wavelet-AR model can detect all the attacks towards to ADSL modem. Wavelet-AR model can detects all the attacks when ADSL modem is in Router mode. Wavelet-AR model can detect most of the attacks when ADSL modem is in Switch mode.

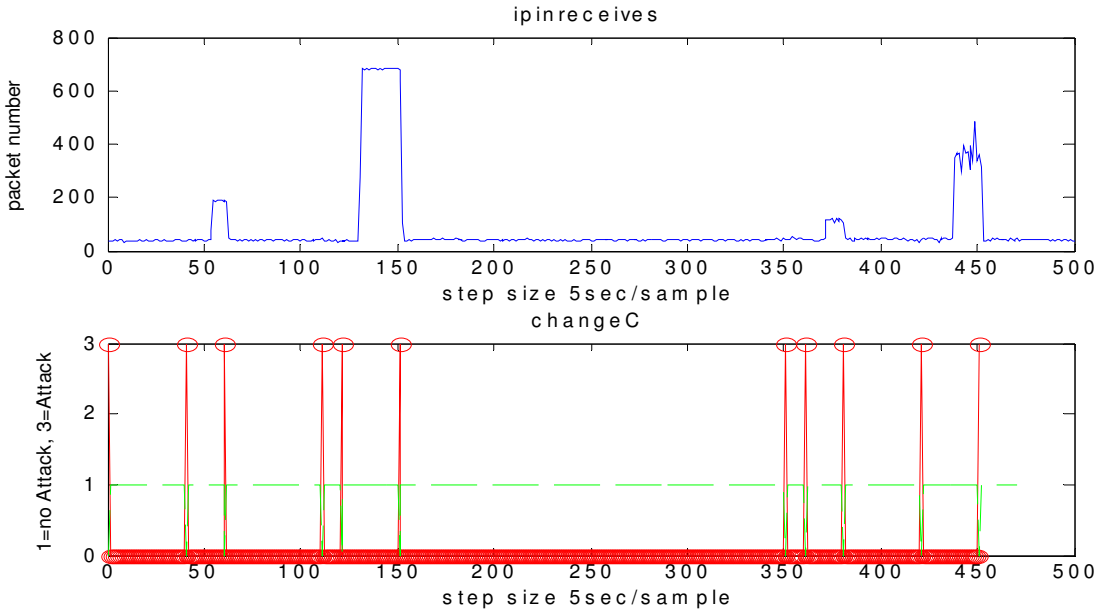

Figure 7.20. Test4_21 Wavelet-AR model decision vector and IPinreceives variables, t=10, 3 input

Table 7.11. Performance measurements of Test4_21

Test4_21, 3 input	**Base Rate**	**Alarms**	**No-Alarms**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	t(sec)
Wavelet- AR model (SVD)	38,00%	16	34	0,064	0,935	0,736	0,263	0,875	0,852	0,390	10
AR Model (SVD)	38,00%	17	33	0,096	0,903	0,736	0,263	0,823	0,848	0,338	10
Wavelet- AR model (SVD)	32,00%	29	71	0,073	0,926	0,750	0,250	0,827	0,887	0,388	5
AR Model (SVD)	32,00%	31	69	0,102	0,897	0,750	0,250	0,774	0,884	0,340	5

Table 7.12. Performance measurements of Test4_21

Test4_21, 4 input	**Base Rate**	**Alarms**	**No-Alarms**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	t(sec)
Wavelet- AR model (SVD)	38,00%	19	31	0,064	0,935	0,894	0,105	0,894	0,935	0,584	10
AR Model (SVD)	38,00%	18	32	0,064	0,935	0,842	0,157	0,888	0,906	0,511	10
Wavelet- AR model (SVD)	32,00%	33	67	0,058	0,941	0,906	0,093	0,878	0,955	0,610	5
AR Model (SVD)	32,00%	34	66	0,102	0,897	0,843	0,156	0,794	0,924	0,444	5

From Table 7.11. and Table 7.12. we can understand that the additional variable LAN increase the detection rate, but it also increase the False alarms. We have face to a trade off between detection rate and false alarms. We have Cid (Intrusion Detection Capability) parameter to decide which mode is more proper. Models are three input or four input modes. If the Cids are almost equal we can choose the one that has greater detection ratio.

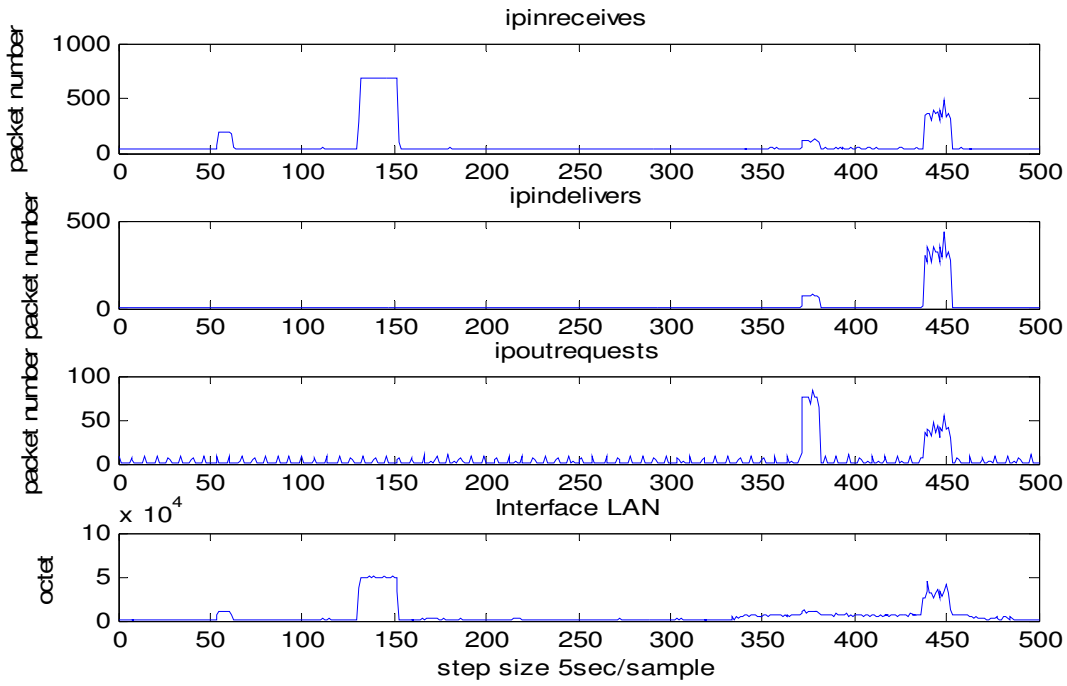

Figure 7.21. Test4_21 data IP variables and interface LAN

Figure 7.21. also shows that LAN variable shows attack peaks regarding the attacks towards to PCs.

7.3.3. Test5_21 Data

Test5_21 data include four attacks, the included attacks are brute force and port scan. The attacks can be seen from IPindelivers and IPoutrequests all towards to ADSL modem in Test5_21 which collected in Dataset-3 Network-2. In this setup ADSL modem was working like Router and Switch. The importance of Test5_21 is all the attacks come from Internet. These attacks were performed from SolarWinds Brute Force and Port Scan attack tools.

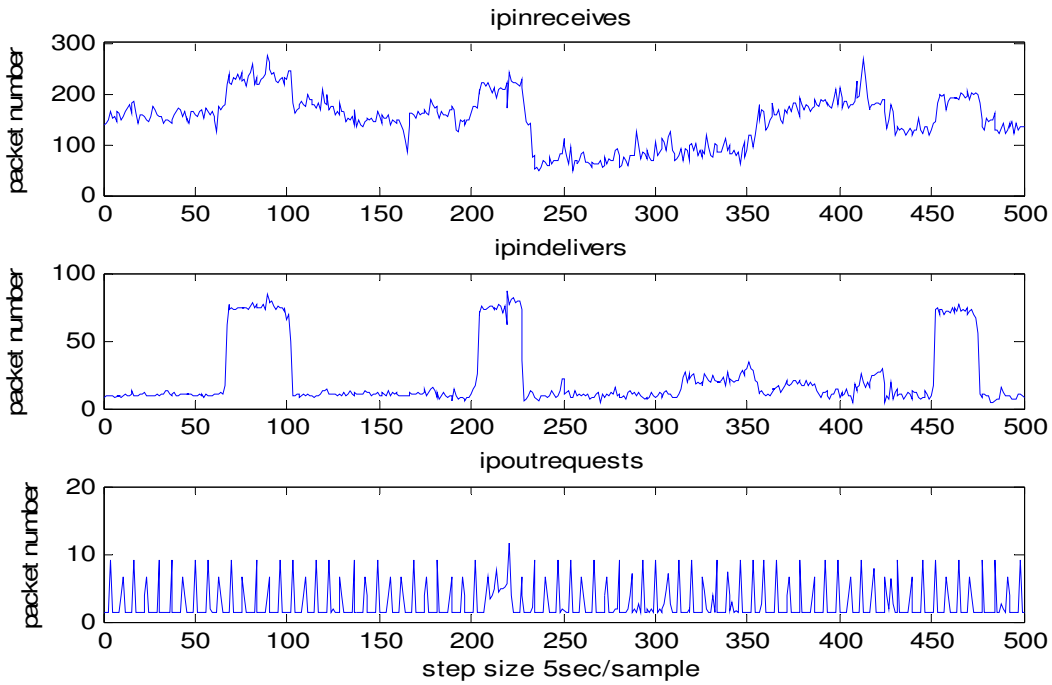

Figure 7.22. Test5_21 data IP variables

As we can see from Figure 7.23. Wavelet-AR model can detect all the attacks towards to ADSL modem which comes from Internet. This setup also shows that Wavelet-AR model can detect attacks which come from internal network or external network (Internet).

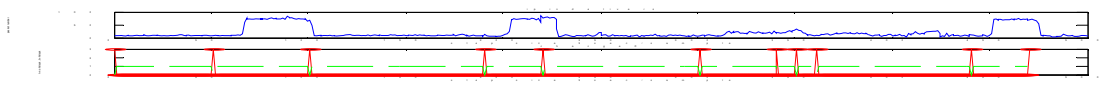

Figure 7.23. Test5_21 decision variable and IPindelivers

7.3.4. Part1 and Part2 Performance Measurements

Part_1 setup includes data Test1_21, Test2_21, Test3_21, and Test4_21.

Table 7.13. Test_Part1_21 performance measurements

Input	Model	I	I'	Base Rate	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid	t (sec)
3 in	W-AR	77	123	0,385	73	127	66	0,057	0,943	0,857	0,143	0,904	0,913	0,546	10
	AR	77	123	0,385	81	119	66	0,122	0,878	0,857	0,143	0,815	0,908	0,434	10
	W-AR	138	262	0,345	134	266	121	0,050	0,950	0,877	0,123	0,903	0,936	0,589	5
	AR	138	262	0,345	141	259	121	0,076	0,924	0,877	0,123	0,858	0,934	0,533	5
	Model	I	I'	Base Rate	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid	t (sec)
4 in	W-AR	77	123	0,385	79	120	69	0,081	0,919	0,896	0,104	0,873	0,934	0,554	10
	AR	77	123	0,385	88	112	69	0,154	0,846	0,896	0,104	0,784	0,929	0,439	10
	W-AR	138	262	0,345	138	262	126	0,046	0,954	0,913	0,087	0,913	0,954	0,653	5
	AR	138	262	0,345	180	220	130	0,191	0,809	0,942	0,058	0,722	0,964	0,454	5

Part_2 setup includes data Test5_21, Test6_21, Test7_21, and Test8_21.

Table 7.14. Test_Part2_21 performance measurements

Input	Model	I	I'	Base Rate	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid	t (sec)
3 in	W-AR	73	127	0,355	61	139	61	0,000	1,000	0,836	0,164	1,000	0,914	0,689	10
	AR	73	127	0,355	71	129	54	0,134	0,866	0,740	0,260	0,761	0,853	0,291	10
	W-AR	138	262	0,345	132	268	122	0,038	0,962	0,884	0,116	0,924	0,940	0,627	5
	AR	138	262	0,345	142	258	120	0,084	0,916	0,870	0,130	0,845	0,930	0,509	5
Input	Model	I	I'	Base Rate	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid	t (sec)
4 in	W-AR	73	127	0,355	71	129	65	0,047	0,953	0,890	0,110	0,915	0,938	0,615	10
	AR	73	127	0,355	72	128	61	0,087	0,913	0,836	0,164	0,847	0,906	0,462	10
	W-AR	138	262	0,345	149	251	125	0,092	0,908	0,906	0,094	0,839	0,948	0,546	5
	AR	138	262	0,345	158	242	126	0,122	0,878	0,913	0,087	0,797	0,950	0,506	5

7.3.5. All Test_21 Performance Measurements

Test_ALL_21 setup includes Test1_21, Test2_21, Test3_21, Test4_21, Test5_21, Test6_21, Test7_21, and Test8_21 data.

Table 7.15. Test_ALL_21 Wavelet-AR model four inputs Performance Measurements

t=10	I	Normal	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid
Test1_21	19	31	21	28	19	0,064	0,935	1,000	0,000	0,904	1,000	0,801
Test2_21	22	28	21	29	20	0,035	0,964	0,909	0,090	0,952	0,931	0,675
Test3_21	17	33	18	32	13	0,151	0,848	0,764	0,235	0,722	0,875	0,292
Test4_21	19	31	19	31	17	0,064	0,935	0,894	0,105	0,894	0,935	0,584
Test5_21	19	31	20	30	19	0,032	0,967	1,000	0,000	0,950	1,000	0,880
Test6_21	17	33	16	34	13	0,090	0,909	0,764	0,235	0,812	0,882	0,374
Test7_21	18	32	17	33	16	0,031	0,969	0,889	0,111	0,941	0,941	0,653
Test8_21	19	31	18	32	17	0,032	0,968	0,895	0,105	0,944	0,938	0,658
Total:	**150**	**250**	**150**	**249**	**134**	**0,064**	**0,936**	**0,893**	**0,107**	**0,893**	**0,936**	**0,583**
Base Rate	**0,375**											

Table 7.16. Test_ALL_21 AR model four inputs Performance Measurements

t=10	I	Normal	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid
Test1_21	19	31	26	24	18	0,258	0,741	0,947	0,052	0,692	0,958	0,391
Test2_21	22	28	20	30	20	0,000	1,000	0,909	0,090	1,000	0,933	0,785
Test3_21	17	33	24	26	15	0,272	0,727	0,882	0,117	0,625	0,923	0,284
Test4_21	19	31	18	32	16	0,064	0,935	0,842	0,157	0,888	0,906	0,511
Test5_21	19	31	18	32	18	0,000	1,000	0,974	0,055	1,000	0,986	0,865
Test6_21	17	33	14	36	12	0,061	0,939	0,706	0,294	0,854	0,861	0,368
Test7_21	18	32	21	29	16	0,156	0,844	0,889	0,111	0,762	0,931	0,424
Test8_21	19	31	19	31	15	0,129	0,871	0,789	0,211	0,789	0,871	0,346
Total:	**150**	**250**	**160**	**240**	**130**	**0,120**	**0,880**	**0,867**	**0,133**	**0,813**	**0,917**	**0,448**
Base Rate	**0,375**											

7.3.6. Comments on Test_21 and Case-2

Brute Force Attacks towards to PCs show traffic changes on IPinreceives and LAN traffic. Brute Force Attacks towards to ADSL Modem show traffic changes on IPindelivers, IPinreceives, IPoutrequests, and LAN traffic.

IPindelivers has lower background noise thus attacks are more significant than other variables. In other words, the range between background noise and attack peak is bigger than other variables, this helps detection. IPoutrequests has low background noise, it shows high traffic changes especially with Port Scan attacks towards to ADSL modem.

Table 7.17. Test_Part1_21 Wavelet-AR versus AR model comparison table

Partitioned															
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	77	123	73	127	66	0,057	0,943	0,857	0,143	0,904	0,913	0,546	10
4 in	W-AR	**Total:**	77	123	79	120	69	0,081	0,919	0,896	0,104	0,873	0,934	0,554	10
3 in	AR	**Total:**	77	123	81	119	66	0,122	0,878	0,857	0,143	0,815	0,908	0,434	10
4 in	AR	**Total:**	77	123	88	112	69	0,154	0,846	0,896	0,104	0,784	0,929	0,439	10
Combined															
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	77	123	73	127	66	0,057	0,943	0,857	0,143	0,904	0,913	0,546	10
4 in	W-AR	**Total:**	77	123	70	130	66	0,032	0,967	0,857	0,143	0,942	0,915	0,602	10
3 in	AR	**Total:**	77	123	77	123	66	0,089	0,915	0,857	0,143	0,857	0,910	0,485	10
4 in	AR	**Total:**	77	123	91	109	68	0,186	0,813	0,883	0,116	0,529	0,966	0,345	10

Table 7.18. Test_Part2_21 Wavelet-AR versus AR model, comparison table

Partitioned															
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	73	127	61	139	61	0,000	1,000	0,836	0,164	1,000	0,914	0,689	10
4 in	W-AR	**Total:**	73	127	71	129	65	0,047	0,953	0,890	0,110	0,915	0,938	0,615	10
3 in	AR	**Total:**	73	127	71	129	54	0,134	0,866	0,740	0,260	0,761	0,853	0,291	10
4 in	AR	**Total:**	73	127	72	128	61	0,087	0,913	0,836	0,164	0,847	0,906	0,462	10
Combined															
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	73	127	92	108	69	0,181	0,819	0,945	0,055	0,750	0,963	0,475	10
4 in	W-AR	**Total:**	73	127	74	126	67	0,055	0,945	0,928	0,082	0,905	0,952	0,640	10
3 in	AR	**Total:**	73	127	90	110	60	0,236	0,764	0,882	0,178	0,667	0,882	0,259	10
4 in	AR	**Total:**	73	127	66	134	62	0,236	0,764	0,882	0,178	0,667	0,882	0,259	10

Table 7.19. Test_ALL_21 Wavelet-AR versus AR model, comparison table

Partitioned															
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	150	250	134	266	127	0,028	0,972	0,847	0,153	0,948	0,914	0,600	10
4 in	W-AR	**Total:**	150	250	150	249	134	0,064	0,936	0,893	0,107	0,893	0,936	0,583	10
3 in	AR	**Total:**	150	250	152	248	120	0,128	0,872	0,800	0,200	0,789	0,879	0,359	10
4 in	AR	**Total:**	150	250	160	240	130	0,120	0,880	0,867	0,133	0,813	0,917	0,448	10
Combined															
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	150	250	131	269	127	0,016	0,984	0,846	0,153	0,969	0,914	0,635	10
4 in	W-AR	**Total:**	150	250	129	271	124	0,020	0,980	0,826	0,173	0,961	0,904	0,596	10
3 in	AR	**Total:**	150	250	134	266	123	0,044	0,956	0,820	0,180	0,917	0,898	0,526	10
4 in	AR	**Total:**	150	250	121	279	115	0,024	0,976	0,766	0,233	0,950	0,874	0,511	10

7.4. CASE-3 ADSL Modem Working as a Router Mode

These data were captured in dataset-4. The network was represented in Figure 5.15. was used for data capturing. Case-3, Dataset-4 includes two types of data. First part of data was gathered under low traffic, second part of was gathered under high traffic volume. First part of data includes Test1_20 and Test2_20 datasets. Second part of data includes Test1_29, Test2_29, Test3_29 and Test4_29 datasets. Case-3 setup was explained in Dataset-4, there were four different networks which are separated from subnet-masks and ADSL modem was working like real router.

7.4.1. Test1_20

Test1_20 data include six attacks, the included attacks are brute force and port scan. The attacks can be seen from IPinreceives all towards to PCs which was in different network. In this setup ADSL modem was working like a Router. The importance of Test1_20 is all the attacks towards to PCs. These attacks were performed from SolarWinds Brute Force and Port Scan attack tools.

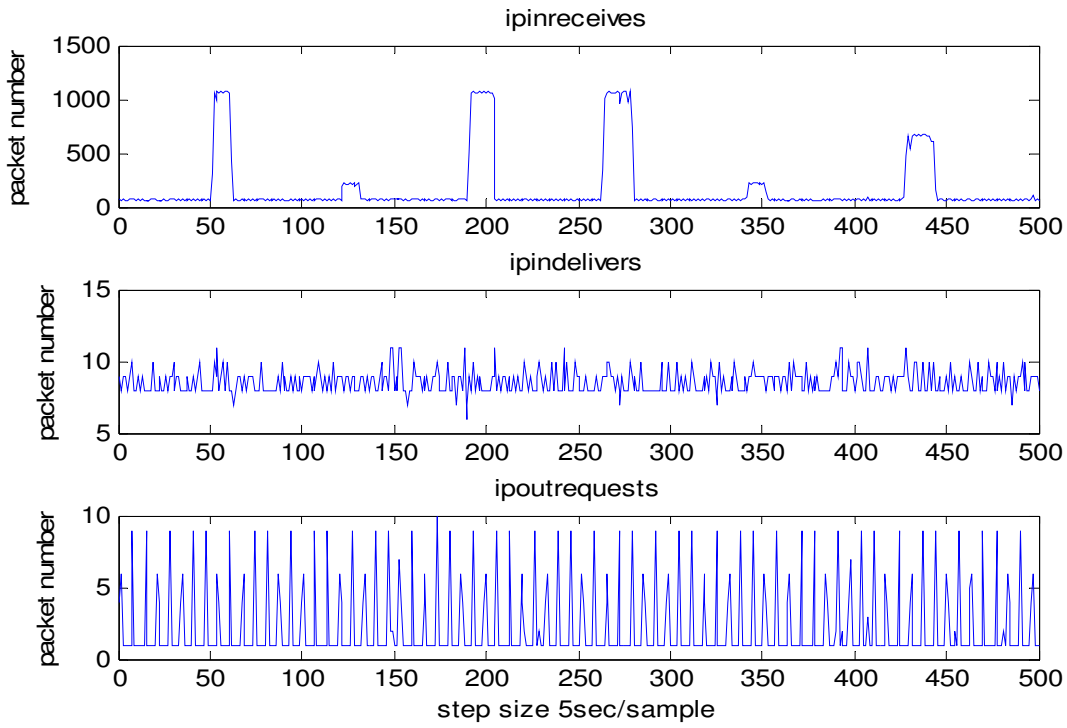

Figure 7.24. Test1_20 IP variables

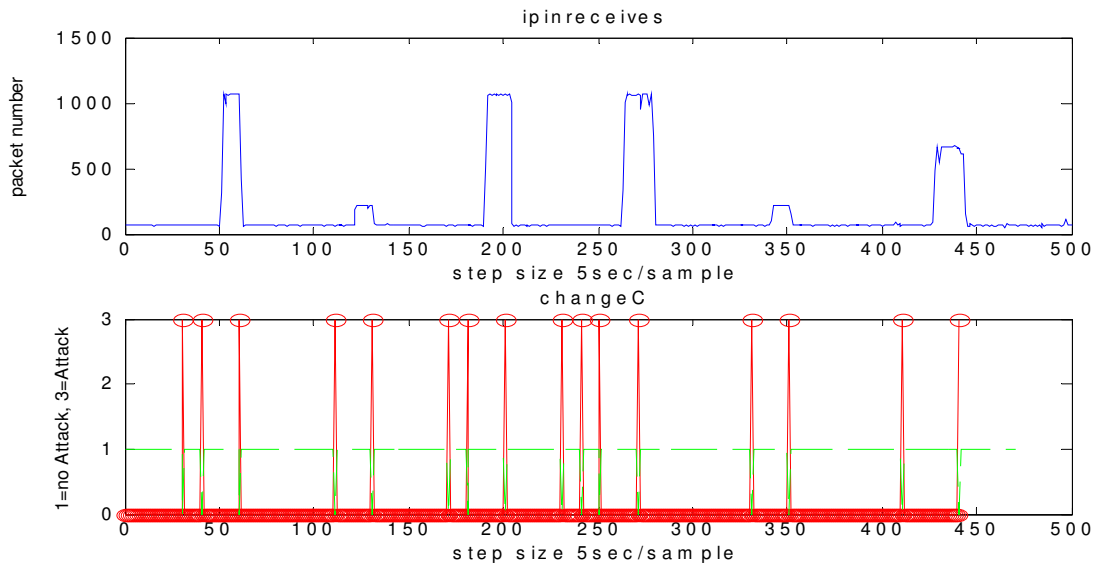

Figure 7.25. Test1_20 decision variable and IPinreceives, t=10, 3 input

As we can see from Figure 7.25. Wavelet-AR model can detect all the attacks towards to PCs which comes from different network. As we can see from Figure 7.24. the attack instances can only be seen from IPinreceives variable. IPindelivers gives information regarding the source node. Since attacks towards the PCs only pass the ADSL modem IPindelivers can not show any changes when attack occurs.

7.4.2. Test_ALL_20 Results

Table 7.20. Test_ALL_20 Wavelet-AR versus AR model comparison table

Input	Model	Data	I	I'	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid	t(sec)
3 in	W-AR	**test1_20**	19	32	23	27	19	0,129	0,871	1,000	0,000	0,826	1,000	0,680	10
3 in	W-AR	**test2_20**	20	30	22	28	20	0,067	0,933	1,000	0,000	0,909	1,000	0,801	10
3 in	W-AR	**Total:**	**39**	**62**	**45**	**55**	**39**	0,098	0,902	1,000	0,000	0,867	1,000	0,736	10
4 in	W-AR	**test1_20**	19	32	21	29	19	0,065	0,935	1,000	0,000	0,905	1,000	0,801	10
4 in	W-AR	**test2_20**	20	30	22	28	20	0,067	0,933	1,000	0,000	0,909	1,000	0,801	10
4 in	W-AR	**Total:**	**39**	**62**	**43**	**57**	**39**	0,066	0,934	1,000	0,000	0,907	1,000	0,801	10
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	AR	**test1_20**	19	32	21	29	12	0,290	0,710	0,632	0,368	0,571	0,759	0,085	10
3 in	AR	**test2_20**	20	30	22	28	20	0,067	0,933	1,000	0,000	0,909	1,000	0,801	10
3 in	AR	**Total:**	**39**	**62**	**43**	**57**	**32**	0,180	0,820	0,821	0,179	0,744	0,877	0,317	10
4 in	AR	**test1_20**	19	32	25	25	16	0,290	0,710	0,842	0,158	0,640	0,880	0,232	10
4 in	AR	**test2_20**	20	30	22	28	17	0,167	0,833	0,850	0,150	0,773	0,893	0,366	10
4 in	AR	**Total:**	**39**	**62**	**47**	**53**	**33**	0,230	0,770	0,846	0,154	0,702	0,887	0,887	10

Table 7.21. Test_ALL_20 Wavelet-AR versus AR model comparison table

						Partitioned									
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	**39**	**62**	**45**	**55**	**39**	0,098	0,902	1,000	0,000	0,867	1,000	0,736	10
4 in	W-AR	**Total:**	**39**	**62**	**43**	**57**	**39**	0,066	0,934	1,000	0,000	0,907	1,000	0,801	10
3 in	AR	**Total:**	**39**	**62**	**43**	**57**	**32**	0,180	0,820	0,821	0,179	0,744	0,877	0,317	10
4 in	AR	**Total:**	**39**	**62**	**47**	**53**	**33**	0,230	0,770	0,846	0,154	0,702	0,887	0,887	10
						Combined									
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	**39**	**62**	**43**	**57**	**39**	0,066	0,934	1,000	0,000	0,907	1,000	0,801	10
4 in	W-AR	**Total:**	**39**	**62**	**42**	**58**	**39**	0,049	0,951	1,000	0,000	0,929	1,000	0,838	10
3 in	AR	**Total:**	**39**	**62**	**38**	**62**	**25**	0,213	0,787	0,641	0,359	0,658	0,774	0,140	10
4 in	AR	**Total:**	**39**	**62**	**48**	**52**	**32**	0,262	0,738	0,821	0,179	0,667	0,865	0,236	10

As we can see from Table 7.20. and Table 7.21. we have high detection rate both in partition mode and combined mode. Partition mode represents the results which taken from data units one by one. In other words Test1_20 and Test2_20 analyzed separately. Combined mode represents the results taken from combined data. In other words Test1_20 and Test2_20 combined and the joint data analyzed together.

7.4.3. Test1_29

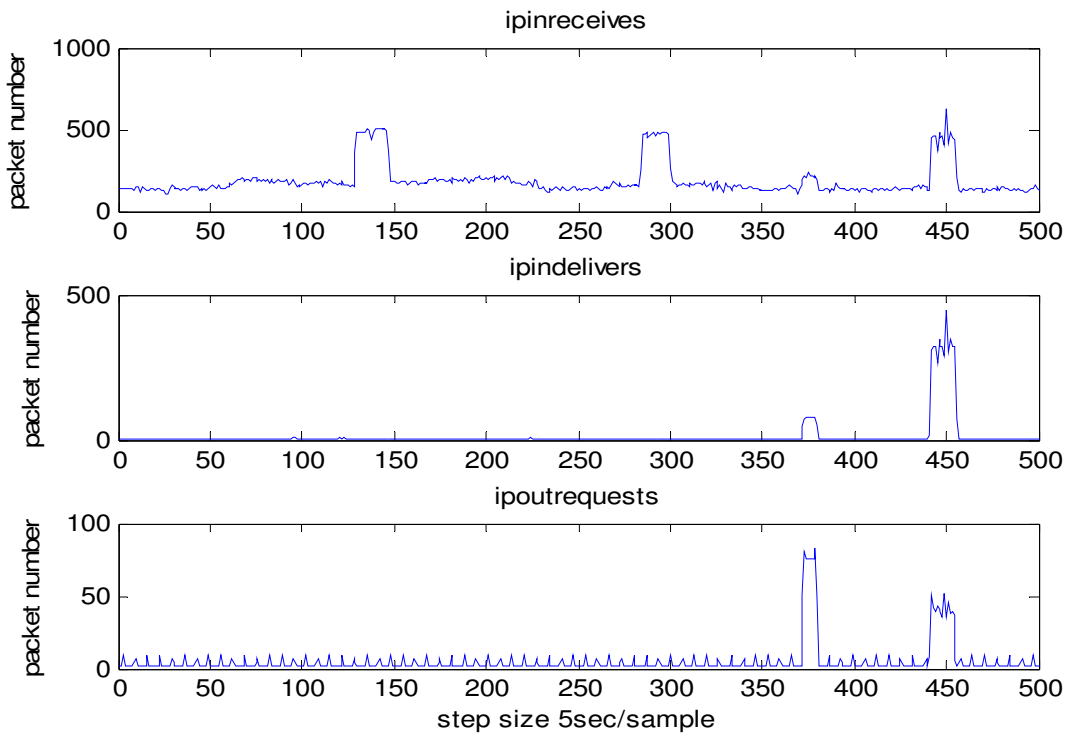

Figure 7.26. Test1_29 IP variables

Test1_29 data include six attacks, the included attacks are brute force and port scan. The attacks can be seen from IPinreceives all towards to PC and ADSL modem. In this setup ADSL modem was working like a Router. The importance of Test1_29 is all PCs were in different networks. These attacks were performed with SolarWinds Brute Force and Port Scan attack tools under high volume traffic.

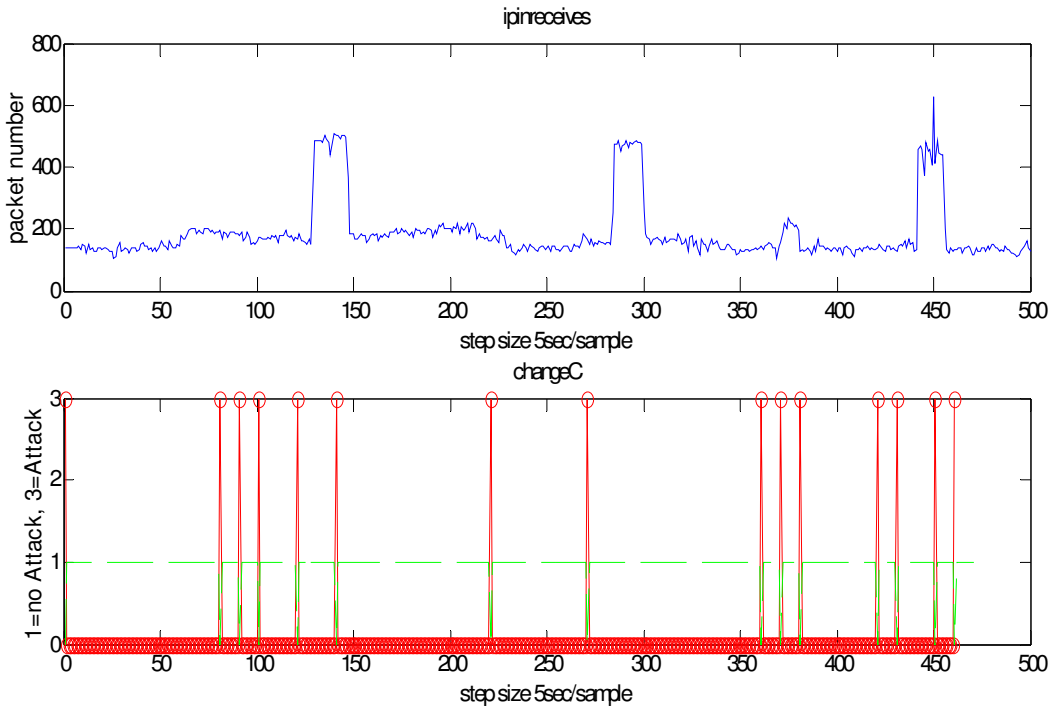

Figure 7.27. Test1_29 decision variables and IPinreceives, t=10, 3 input

There are some attacks can not be seen from IP variables, this because of the high traffic volume. Especially port scan attacks to PCs can not be easily seen from IP variables. Although some of the attacks can not be easily seen from IP variables Wavelt-AR model can detect most of them.

7.4.4. Test_ALL_29 Results

There were four pieces of test data, their names are Test1_29, Test2_29, Test3_29 and Test4_29 we have added some of the results tables, all other tables will be put in Appendix.

Table 7.22. Test_ALL_29 Wavelet-AR performance table, 3 input

t=10	I	Normal	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid
Test1_29	21	29	24	26	21	0,103	0,897	1,000	0,000	0,875	1,000	0,734
Test2_29	18	32	19	31	16	0,094	0,906	0,889	0,111	0,842	0,935	0,519
Test3_29	21	29	23	27	21	0,069	0,931	1,000	0,000	0,913	1,000	0,800
Test4_29	18	32	19	32	16	0,094	0,906	0,889	0,111	0,842	0,935	0,519
Total:	**78**	**122**	**85**	**116**	**74**	**0,09**	**0,91**	**0,95**	**0,05**	**0,871**	**0,965**	**0,625**
Base Rate												

t=5	I	Normal	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid
Test1_29	40	60	43	57	40	0,050	0,950	1,000	0,000	0,930	1,000	0,838
Test2_29	35	65	43	57	35	0,123	0,877	1,000	0,000	0,814	1,000	0,681
Test3_29	40	60	46	54	40	0,100	0,900	1,000	0,000	0,870	1,000	0,735
Test4_29	35	65	41	59	31	0,154	0,846	0,886	0,114	0,756	0,932	0,422
Total:	**150**	**250**	**173**	**227**	**146**	**0,108**	**0,892**	**0,973**	**0,027**	**0,844**	**0,982**	**0,641**
Base Rate												

Table 7.23. Test_ALL_29 AR performance table, 3 input

t=10	I	Normal	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid
Test1_29	21	29	26	24	18	0,276	0,724	0,857	0,143	0,692	0,875	0,262
Test2_29	18	32	18	32	15	0,094	0,906	0,833	0,167	0,833	0,906	0,447
Test3_29	21	29	30	20	18	0,414	0,586	0,857	0,143	0,600	0,850	0,158
Test4_29	18	32	20	30	12	0,250	0,750	0,667	0,333	0,600	0,800	0,129
Total:	**78**	**122**	**94**	**106**	**63**	**0,254**	**0,746**	**0,808**	**0,192**	**0,670**	**0,858**	**0,231**
Base Rate	**0,385**											

t=5	I	Normal	Alarm	Normal	Detected	FP	TN	TP	FN	PPV	NPV	Cid
Test1_21	40	60	46	54	40	0,100	0,900	1,000	0,000	0,870	1,000	0,740
Test2_21	35	65	37	63	30	0,108	0,892	0,857	0,143	0,811	0,921	0,453
Test3_21	40	60	56	44	40	0,267	0,733	1,000	0,000	0,714	1,000	0,502
Test4_21	35	65	48	52	32	0,246	0,754	0,914	0,086	0,667	0,942	0,351
Total:	**150**	**250**	**187**	**213**	**142**	**0,180**	**0,820**	**0,947**	**0,053**	**0,759**	**0,962**	**0,481**
Base Rate	**0,345**											

As we can see from Table 7.22. and Table 7.23. we have high detection rate.

7.4.5. Comments on Test_ALL_29 Results

First part of data was captured under low traffic, and detection rate was really high in this setup, True positive is one in all Wavelet-AR analysis. Second part of was captured under high traffic and detection rate is also high in this setup, but detection rate is decrease 3 or 5 per cent.

There are some attacks can not be seen from IP variables, this because of the high traffic volume. Especially Port Scan attacks to PCs can not be easily seen from IP variables. Although some of the attacks can not be easily seen from IP variables Wavelt-AR model can detect most of them. This also shows the detection sensitivity of the AR analysis.

Table 7.24. Test_ALL_29 Wavelet-AR versus AR comparison table

						Partitioned									
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	**78**	**122**	**85**	**116**	**74**	0,090	0,910	0,949	0,051	0,871	0,965	0,625	10
4 in	W-AR	**Total:**	**78**	**122**	**86**	**115**	**74**	0,098	0,902	0,949	0,051	0,860	0,965	0,611	10
3 in	AR	**Total:**	**78**	**122**	**94**	**106**	**63**	0,254	0,746	0,808	0,192	0,670	0,858	0,231	10
4 in	AR	**Total:**	**78**	**122**	**94**	**117**	**69**	0,205	0,795	0,885	0,115	0,734	0,915	0,363	10
						Combined									
Input	**Model**	**Data**	**I**	**I'**	**Alarm**	**Normal**	**Detected**	**FP**	**TN**	**TP**	**FN**	**PPV**	**NPV**	**Cid**	**t(sec)**
3 in	W-AR	**Total:**	**78**	**122**	**87**	**113**	**71**	0,131	0,869	0,910	0,090	0,816	0,938	0,493	10
4 in	W-AR	**Total:**	**78**	**122**	**86**	**114**	**71**	0,123	0,877	0,910	0,090	0,826	0,939	0,506	10
3 in	AR	**Total:**	**78**	**122**	**82**	**118**	**55**	0,221	0,779	0,705	0,295	0,671	0,805	0,176	10
4 in	AR	**Total:**	**78**	**122**	**80**	**120**	**63**	0,139	0,861	0,808	0,192	0,788	0,875	0,353	10

8. CONCLUSION

We have proposed a new *A* operator matrix definition with Singular Value Decomposition (SVD) method. We have tested Yule-Walker and Modified Yule-Walker instead of Least Squares in [1]. We have shown that Yule-Walker and Modified Yule-Walker have better performance than Least-Squares and they are more reliable than Least Squares. We think that we could improve the *Totthan et al's* Least Squares Autoregressive based IDS via a new A operator matrix based on SVD method and using Yule-Walker parameter estimation.

We have done some performance test with Wavelet-Modulus Maxima. This method has disadvantages; it could not detect attacks which do not have significant traffic volume changes in whole traffic data. It could easily detect the attacks which have significant changes in Ethernet traffic, but these attacks have different behavior in whole traffic data. Wavelet-Modulus Maxima model could not detect hidden attacks.

We have done performance test with new Log Likelihood Ration LLR (η_L), but we saw that new LLR has low detection rate. In other words newly proposed LLR is not sensitive as firstly proposed η.

Our actual contribution is to combine the Wavelet and Autoregressive models to reach a reliable Intrusion Detection System. We have proposed a combined Wavelet and AR intrusion detection system. The Wavelet-AR is designed to satisfy common performance criterion of Intrusion Detection Systems. The common performance criterions are True Positive Rate (TP), True Negative Rate (TN), False Positive Rate (FP), False Negative Rate (FN), Positive Predictive Value (PPV), Negative Predictive Value (NPV), and Intrusion Detection Capability (C_{ID}). We have shown in performance tables that we could improve the AR model's performance, in other words we make the IDS more reliable. We could decrease the FP rate via using Wavelet Decomposition before the AR model. We have used only approximate coefficients of wavelet decomposition this yield a false alarm reduction. This reduction is because of wavelet transform properties. Wavelet transform is eliminating the high variations from data; these high variations can cause a

false alarm in AR model because of its sensitivity. Since we have used approximate coefficients of wavelet transform of data. Approximate coefficients of wavelet transform of data gives us smoother version of actual data so we could applied AR model on approximate coefficients. And also we have shown that wavelet transform did not reduce the detection rate because of AR analysis is very sensitive.

Performance measurements have shown that Wavelet-AR model with shift length t=5sec has the best results and highest C_{ID}. In other words shift length t=5sec has the best detection rate with acceptable False Positive rate. We have used firstly proposed likelihood ratio which has better detection rate.

The performance measurements have showed that, the proposed method performs well in detecting some traffic-related attacks such as DoS and DDoS. The results also show that the method can play an important role in the integration of IDS and network management system.

We could combine the Wavelet and Autoregressive models and we could get better performance with Wavelet-AR model than AR model as shown in Performance Measurements Tables.

In order to understand the behavior of Wavelet-AR and AR model, we have performed several tests. They are Case-2, Case-3, and Case-4. Case-2 includes nine pieces of tests, Case-3 includes eight pieces of tests, and Case-4 includes six pieces of test. In these tests ADSL modem has worked as a Switch, Switch and Router, and Router mode at Case-2, Case-3, and Case-4 setups respectively.

Wavelet-AR model can detect all the attacks towards to ADSL modem which come from internal network, but it can not detect attacks towards to PC2 and PC3 while ADSL modem was working like a switch. We have added the variable *Ifinoctet* to Wavelet-AR model. The analysis done with four inputs (IP variables and *Ifinoctet* LAN variable) has better True Positive values than the analysis done with three inputs (only IP variables). This detection rate increase comes from variable LAN. We can fairly say that LAN

variable has increased the model detection rate when ADSL modem was working in switch mode.

Brute Force Attacks towards to PCs show traffic changes on IPinreceives and LAN traffic. While ADSL modem was working like a Router. Brute Force Attacks towards to ADSL modem show traffic changes on IPindelivers, IPinreceives, IPoutrequests, LAN traffic in all cases.

IPindelivers has lower background noise thus attacks are more significant than other variables. In other words, the range between background noise and attack peak is bigger than other variables. *IPoutrequests* has low background noise also, but it shows high traffic changes especially with Port Scan attacks towards to ADSL modem.

True positive value is one in all Wavelet-AR analysis when the ADSL modem was working like a Router and data captured under low traffic. When data captured under high traffic, detection rate is also high, but detection rate has decreased 3 or 5 per cent.

If we take samples with 1Gbitps and take 1000.000 samples with 1000 samples per second, 1000.000 samples equals to 1000 second and we are doing 232.140 Kilo FLOP operations. In this case number of FLOP operations is 13.78 times smaller than P4 processor FLOP operations. We can reduce the window size to 100 second and we can increase the data rate up to 10Gbitps. Wavelet-AR model can work real-time with 10Gbitps data rate with 100 second observation window with P4 standard Laptop.

While we found the results to be encouraging, we are aware of that these were limited experiments, on a local test environment, under controlled traffic loads. Future work is needed to validate the method under more realistic and complicated traffic conditions. Future efforts will be also focused on anomaly detection on intrusion different attacks.

Consequently, we have observed that we had achieved the design objectives of an Intrusion Detection System. Wavelet-AR may be treated as a reliable and robust Intrusion Detection System to be used for traffic based intrusion detection.

APPENDIX A: PERFORMANCE TABLES

The Appendix A has been given as a separate CD which includes;

- CASE-1_Dataset-2 Folder, includes; (you need Microsoft Office Excel)

  - Dataset-2_attack_instances_performance-Tables.xls
  - Dataset-2_performance_analizer and Tables.xls

- CASE-2_Dataset-3 Folder, includes (you need Microsoft Office Excel)

  - attack_instances_performance_part_1_Tables.xls
  - attack_instances_performance_part_2_Tables.xls
  - performance_analizer_part_1_Tables.xls
  - performance_analizer_part_2_Tables.xls
  - performance_analizer_ALL_21_Tables.xls
  - performance_ALL_Combined_21_Tables.xls

- CASE-3_Dataset-4 Folder, includes (you need Microsoft Office Excel)

  - attack_instances_performance_test20_Tables.xls
  - performance_analizer_test20_Tables.xls
  - attack_instances_performance_test29_Tables.xls
  - performance_analizer_29_Tables.xls

- Appendix_A.doc (you need Microsoft Office Word or Wordpad)

  Includes ALL Performance Measurements Tables.

APPENDIX B : MATLAB CODES

In this part we give the simulation codes of AR model, Wavelet-Modulus Maxima, and Wavelet-AR. The implementation is evaluated in MATLAB R14 environment. We choose MATLAB as it provides efficient computations on vectors and matrices. Below we have listed the MATLAB functions, we have write all function in detailed. We have listed main codec in italic and we have listed with numbers required function of main codes.

Main codes:

- wavelet_mTotthan_dec_ver4.m: AR Model main code, 3 input
- wavelet_mTotthan_dec_ver4_1.m: AR Model main code, 4 input
- wavelet_modmax_perdev1.m: Wavelet Modulus Maxima Model main code
- waveletdb6_mTotthan.m: Wavelet-AR model main code
- wavelet_mTotthan_dec_ver3.m: Wavelet-AR model main code, 3 input
- wavelet_mTotthan_dec_ver3.m: Wavelet-AR model main code, 4 input
- e_ar.m : Least-Squares AR main code

Functions:

1. decision.m: Percentage Deviation Decision
2. decision1.m: Percentage Deviation Decision
3. decision2.m: Percentage Deviation Decision
4. eyule_wavelet_ext.m :A code call for Yule Walker
5. eyule_wavelet_ext1.m : A code call for Yule Walker
6. eyule_wavelet_ext2.m : A code call for Yule Walker
7. eyule_wavelet_ext3.m : A code Modified Yule-Walker
8. eyule_wavelet_ext4.m : A code call for Yule Walker
9. eyule_wavelet_ver3.m : A code call for Modified Yule Walker
10. modulusmaxima.m: Modulus Maxima calculation
11. mywarma.m : Modified Yule Walker by R. Moses

12. lnmywarma.m: A code call for Modified Yule Walker
13. lnar.m: A code call for Least-Squares
14. lsar.m Least Squares by R. Moses
15. perdeviation.m : Percentage Deviation Calculation
16. perdeviation1.m : Percentage Deviation Calculation
17. vectordivider.m: Vector divider for less CPU usage
18. yule_ar.m : A code call for Yule Walker
19. yulewalker.m: Yule Walker by R. Moses
20. yule_ar_ver1.m : Advance Analysis on vector via Yule-Walker
21. yule_ar_ver2.m : Advance Analysis on vector via Yule-Walker
22. ppv_cid_tester.m
23. ppv_npv.m
24. cid.m

Matlab R14 Functions are necessary for main codes

- wavedec.m
- wrcoef.m
- svd.m

Table B.1. Data were used for analyses

Data	**Dataset**	**Name**
snmp_data.mat	Dataset-1	Data with Attacks-1
olddata.mat	Dataset-1	Data with Attacks-1
data2.mat	Dataset-1	Attack Free Data
a1b1c1.mat	Dataset-1	Artificial Data
test2_02.mat	Dataset-2	Test2_02
test3_02_all.mat	Dataset-2	Test3_02
test1_03_all.mat	Dataset-2	Test1_03
test2_03.mat	Dataset-2	Test2_03
test4_03_all.mat	Dataset-2	Test4_03
test6_03_all.mat	Dataset-2	Test6_03
test7_03_all.mat	Dataset-2	Test7_03
test8_03_all.mat	Dataset-2	Test8_03
test9_03_all.mat	Dataset-2	Test9_03
test_all_03.mat	Dataset-2	Test_ALL_0203
test_all_lan.mat	Dataset-2	Test_ALL_0203

Data	Continued Dataset	Name
test1_21.mat	Dataset-3	Test1_21
test2_21.mat	Dataset-3	Test2_21
test3_21.mat	Dataset-3	Test3_21
test4_21.mat	Dataset-3	Test4_21
test5_21.mat	Dataset-3	Test5_21
test6_21.mat	Dataset-3	Test6_21
test7_21.mat	Dataset-3	Test7_21
test8_21.mat	Dataset-3	Test8_21
test_part1_21.mat	Dataset-3	Test_part1_21
test_part1_21.mat	Dataset-3	Test_part2_21
test_all_21.mat	Dataset-3	Test_ALL_21
test1_20.mat	Dataset-4	Test1_20
test2_20.mat	Dataset-4	Test2_20
test_all_20.mat	Dataset-4	Test_ALL_20
test1_29.mat	Dataset-4	Test1_29
test2_29.mat	Dataset-4	Test2_29
test3_29.mat	Dataset-4	Test3_29
test4_29.mat	Dataset-4	Test4_29
test_all_29.mat	Dataset-4	Test_ALL_29

The Appendix B has been given as a separate CD includes;

- Appendix B.doc (you need Microsoft Office Word or Wordpad)
  Includes matlab codes definition and how to run them.

- Appendix B Tables.xls (you need Microsoft Office Excel)
  Includes the main code's functions.

- wavelet summary_1.doc (you need Microsoft Office Word or Wordpad)
  Includes codes definitions and their functions.

- Matlab Codes and Data Folder (You need Matlab R14, Wavelet Toolbox or wavedec.m and wrcoef.m Matlab functions)
  Includes all Matlab Functions and data used for analyses.
  All the included files and data were explained in Appendix_B_.doc

REFERENCES

1. Chuanyi Ji, Marina Thottan, “Anomaly detection in IP Networks”, *IEEE Transactions on Signal Processing*, 51(8):2191-2204, 2003.

2. M. Thottan and C. Ji, “Proactive anomaly detection using distributed. Intelligent agents” *IEEE Network,* vol. 12, pp. 21–27, Sept./Oct. 1998

3. M. Thottan and C. Ji, "Adaptive thresholdding for proactive network problem detection", *Proc. IEEE Int. Workshop Syst. Manage.,* Newport, RI, Apr. 1998.

4. M. Thottan and C. Ji, *Fault Prediction at the Network Layer Using Intelligent Agents* www.bell-labs.com/user/marinat/pubs/im99.pdf

5. PETER De SOUZA, "Statistical Test and Distance Measures for LPC Coefficients” *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. Assp-25 No:6, December 1977

6. Stoica, P., and R. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice Hall, 1997 pp. 89-90, 103-106,

7. *SVD Method*, http://en.wikipedia.org/wiki/Singular_value_decomposition

8. *Correlation Coefficients*, http://en.wikipedia.org/wiki/Correlation_coefficient

9. *Medain*, http://en.wikipedia.org/wiki/Median

10. H.J.Barnard, *Image and Video Coding Using a Wavelet Decomposition*, Ph.D. Thesis, Delft University, ISBN 90-5326-016-1, pp. 14-21, 1994.

11. Donald B. Percival, *Wavelet Methods for Time Series Analysis*, Cambrige University Press 2000. pp. 1, 2, 5-9, 13, 17,

12. Matlab, “wavedec” function, “Matlab R14 Wavelet Toolbox”

13. Stephane Mallat and Wen Liang Hwang. “Singularity Detection and Processing with Wavelets”, *IEEE Transactions on Information Theory*, Vol.38, No.2, March 1992.

14. Guofei gu, Prahlad Fogla, David Dagon, Wenke Lee, *An Information-Theoretic Measure of Intrusion Detection Capability*, http://smartech.gatech.edu/dspace/bitstream/1853/9432/1/GIT-CC-05-10.pdf

15. Jo˜ao B. D. Cabrera_ , Lundy Lewis_ , Xinzhou Qin_ ,Wenke Lee_ , Ravi K. Prasanth_ , B. Ravichandran_ and Raman K. Mehra, “Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables -A Feasibility Study”, *Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management,* Seattle, WA - May 14-18, 2001.

16. Damiano Bolzoni, Sandro Etalle, *APHRODITE: an Anomaly-based Architecture for False Positive Reduction*, http://arxiv.org/PS_cache/cs/pdf/0604/0604026.pdf

17. A. C. Gilbert, “Multiscale Analysis and Data Networks”, *Applied and computational Harmonic Analysis 10, 185–202 (2001)*

18. Moad Alhamaty, Ali Yazdian, Fathi Al-qadasi, “Intrusion Detection System Based on The Integrity of TCP Packet” *Transactions on Engineering, Computing and Technology*, V11 February 2006 ISSN 1305-5313

19. Y. Zhang, Z. Ge, M. Roughan, and A. Greenberg. “Network Anomography”, *Proceedings of the Internet Measurement Conference (IMC '05)*, Berkeley, CA, USA, October 2005.

20. Chin-Tser Huang, Sachin Thareja, Yong-June Shin, *Wavelet-based Real Time Detection of Network Traffic Anomalies,*http://www.cse.sc.edu/~huangct/wens06.pdf

21. Sanjay Rawat, and Challa S. Sastry, *Network Intrusion Detection Using Wavelet Analysis,* https://www.springerlink.com/content/1ac9arcw4y1bkh3k/resource-secured/?target=fulltext.pdf

22. Vladimir Gudkov and Joseph E. Johnson, *Multidimensional Network Monitoring for Intrusion Detection*, http://arxiv.org/PS_cache/cs/pdf/0206/0206020.pdf

23. Seong Soo Kim, A. L. Narasimha Reddy, and Marina Vannucci, *Detecting Traffic Anomalies through aggregate analysis of packet header data,* http://www.ee.tamu.edu/~reddy/papers/skim_net04.pdf

24. Seong Soo Kim, A. L. Narasimha Reddy, and Marina Vannucci, *Detecting Traffic Anomalies using Discrete Wavelet Transform,* http://www.ece.tamu.edu/~reddy/papers/icoin_skim03.pdf

25. Seong Soo Kim and A. L. Narasimha Reddy, "Image-based Anomaly Detection Technique: Algorithm, Implementation and Effectiveness", *IEEE Journal on Selected Areas in Communications – High-Speed Network Security*, Oct. 2006Volume: 24,

26. Haining Wang, Danlu Zhang, Member, and Kang G. Shin, "Change-Point Monitoring for the Detection of DoS Attacks", *IEEE Transactions on Dependable and Secure Computing,* Vol. 1, No. 4, October –December 2004

27. Khaled Labib and V. Rao Vemuri, *Detecting And Visualizing Denial-of-Service And Network Probe Attacks Using PrincipalComponent Analysis*, http://citeseer.ist.psu.edu/cache/papers/cs2/20/http:zSzzSzailab.das.ucdavis.eduzSz~klabibzSzdocszSzSAR04_Labib_Vemuri_CR_Final.pdf/labib04detecting.pdf

28. Damiano Bolzoni, Sandro Etalle, Pieter Hartel, *POSEIDON: a 2-tier Anomaly-based Network Intrusion Detection System,* http://arxiv.org/PS_cache/cs/pdf/0511/0511043.pdf

29. Joseph Lo, *Denial of Service or Nuke Attacks,* http://www.irchelp.org/irchelp/security/

30. Przemyslaw Kazienko & Piotr Dorosz, *Intrusion Detection Systems (IDS) Part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture).* Apr 07, 2003 www.windowsecurity.com

31. *SNMP*, http://en.wikipedia.org/wiki/SNMP

32. Qingtao Wu, Zhiqing Shao, "Network Anomaly Detection Using Time Series Analysis", *Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services* (ICAS/ICNS 2005) 0-7695-2450-8/05 $20.00 © 2005 IEEE

33. Mehmet Özgür Depren , *Network Based Intelligent Intrusion Detection System*, Master Thesis, Boğaziçi University, 2003. pp. 7-11

34. Orçun Uzun , *A Dynamically Clock-Controlled Binary Stream Cipher with Memory and Dynamic Multiplexing*, Master Thesis, Boğaziçi University,

REFERENCES NOT CITED

Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. "On the Self Similar Nature of Ethernet Traffic (Extended Version)", *IEEE/ACM Transactions on Networking,* Vol.2, No. 1, February 1994.

Stefan Axelsson, *Intrusion Detection Systems: A Survey and Taxonomy*, http://www.mnlab.cs.depaul.edu/seminar/spr2003/IDSSurvey.pdf

Rebecca Bace and Peter Mell, "Intrusion Detection Systems", *NIST Special Publication on Intrusion Detection Systems*: http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf

Nong Ye, Xiangyang Li, Qiang Chen, Syed Masum Emran, and Mingming Xu, "Probabilistic Techniques for Intrusion Detection Based on Computer Audit Data", *IEEE Transactions on Systems, Man, Cubernatics-Part A:systems and Humans*, Vol. 31, No. 4, July 2001

Peyman Kabiri and Ali A. Ghorbani, "Research on Intrusion Detection and Response: A Survey *International Journal of Network Security*, Vol.1, No.2, PP.84–102, Sep. 2005

Christopher Torrence and Gilbert P. Compo, *A Practical Guide to Wavelet Analysis* http://pangea.stanford.edu/research/Oceans/GES290/Torrence1998.pdf

Patrice Abry and Darryl Veitch, "Wavelet Analysis of Long-Range-Dependent Traffic", *IEEE Transactions on Information Theory*, Vol. 44, No. 1, January 1998

Paul Barford, Jeffery Kline, David Plonka and Amos Ron, "A Signal Analysis of Network Traffic Anomalies", *Proceedings of ACM Sigcomm Internet Measurements Workshop,* 2002

Zheng Zhang, C. Manikopoulos, Jay Jorgenson, Jose Ucles, *Comparison of Wavelet Compression Algorithms in Network Intrusion Detection*, http://web.njit.edu/~manikopo/papers/published/ICCIT01Zheng-final.pdf